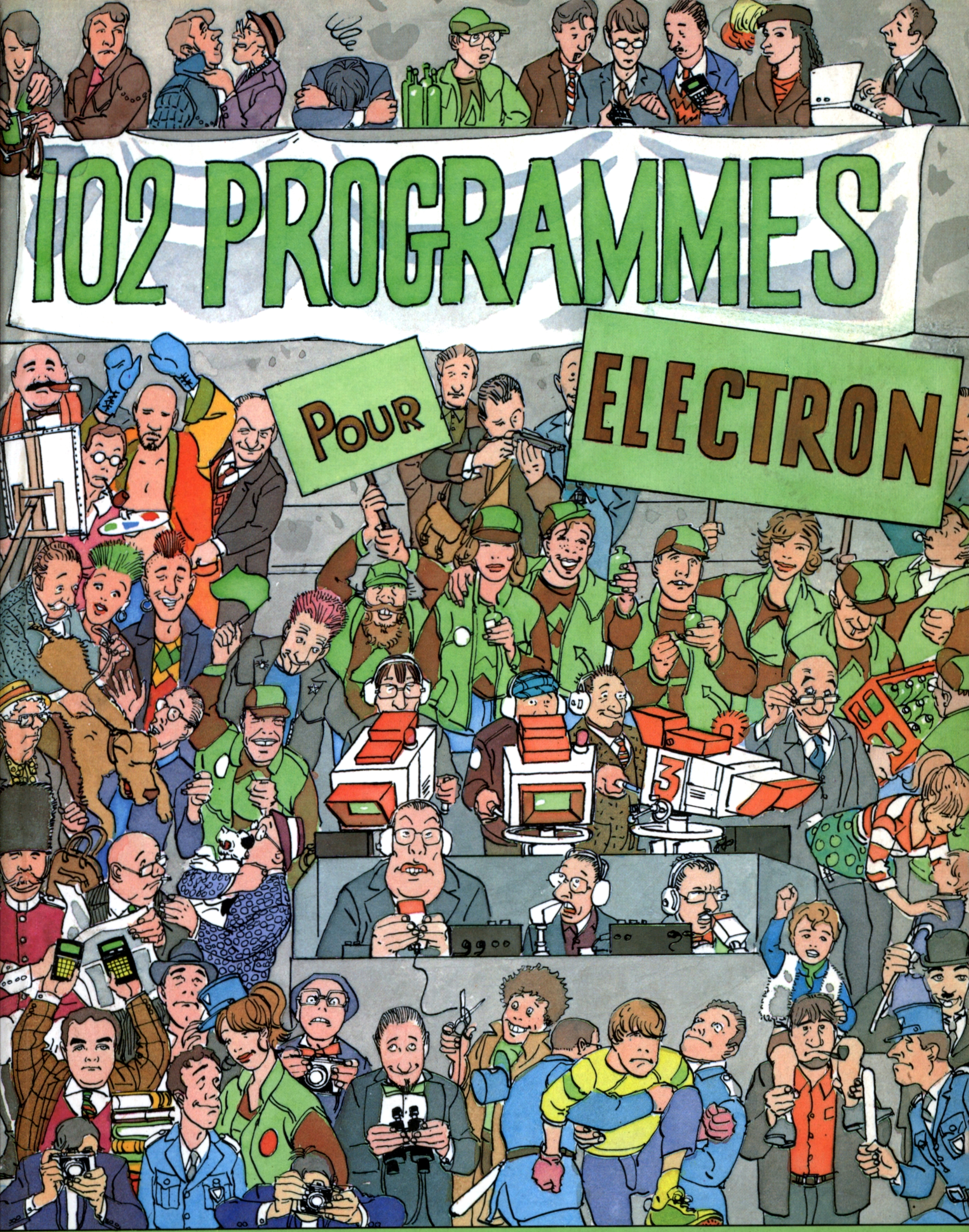


PAR
JACQUES
DECONCHAT

102 PROGRAMMES POUR

ELECTRON

Ouvrages généraux relatifs à la programmation

— Comment programmer - Jean-Claude Barbance.
— Méthodes de réalisation des programmes - Michel Benelfoul.

Ouvrages d'initiation

— Visa pour l'informatique - Jean-Michel Jégo.
— Visa pour le Basic - Jean-Michel Jégo.
— Mon ordinateur - Jean-Claude Barbance.
— Le logotron informatique - Jean-Pierre Petit.
— L'ordinateur apprivoisé - François Picard et Danielle Shaw.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Éditions du P.S.I., B.P. 86, 77402 Lagny Cedex, 1985

ISBN : 2-86595-212-6

JACQUES
DECONCHAT

102 PROGRAMMES POUR ELECTRON

adapté par Jacques LABIDURIE

EDITIONS DU P.S.I.
1985

SOMMAIRE

Avant-propos 8
Organisation de l'ouvrage 9

Niveau 1

Instructions élémentaires 13
Le nombre mystérieux 18
Cube 20
Chaud, froid 22
Les allumettes 24
Allumettes à plusieurs 26
Calendrier perpétuel 28
Bataille navale 30
Golf 32
Solfège 34
Le jeu du 21 36
Pair - Impair 38
Devine un chiffre 40
Le jeu de Wytthof 42
Rebonds 44
Roulette 46
Couloirs 48
Le jeu de la fève 50
Jeu de l'oie aléatoire 52
Jackpot 54

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne 58
Dé truqué 60
Visions 62
421 64
L'électricien fou 66
La course au 2Ø 68
Musique 70
Poker 72
Le jeu de saute-mouton 74
Lettres 76
Espion 78
Accrochez les wagons 80
Tirage d'un dé truqué 82
Tour de cartes 84

Le jeu de Marienbad 86
L'indésirable 88
Compose 90
Jour de la semaine 92
Le jeu de la vie 94

Niveau 3

Manipulation des chaînes - L'instruction DATA 98
Machinamot 100
Un dé 102
Plus grand, plus petit 104
Ordre 106
Le pendu 108
Belotte 110
Taquin de calcul 112
Pendu inversé 114
Nicomaque 116
Course de chevaux 118
A travers 120
Mastermind 122
Idem 124
Horloge 1 126
Reines sur l'échiquier 128
Répétitions 130
Mots 132
Permutations 134
Bridge 136
Tours de Hanoï 138
Le loup et les agneaux 140

Niveau 4

Les techniques d'animation élémentaire 144
Losanges 146
Mathieu 148
Anagrammes 150
Tirage de dés 152
Jeu de loto 154
La bombe 156
Ardoise magique 158
Affichage publicitaire 160
Evasion 162
Combat aérien 164
Dessin de dés 166
Le jeu du missile 168
Le singe 170

Le jeu du 15 172
Adresse 174
Sous-marin 176
Diago 178
Décomaque 180
Clavier 182
Clavier bis 184

Niveau 5

Les techniques spécifiques de l'Electron 189
Smokey 192
Course de voitures 194
Danger spatial 196
Tic Tac Toe 198
Horloge 2 200
Dames 203
Œufs 206
Taquin de lettres 208
Quatre à quatre 211
Cibles 214
Gouttes 216
Obstacles 218
Captures dans l'espace 220
Anomalie 222
Plusieurs en un 224
Parcours du cavalier 226
Mur de briques 228
Echecs 230
Calculatrice 233
Echiquier 236
Zombies 238
Le lézard d'Alice 240
Chapeaux 242
Othello 244

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic, pour l'ordinateur individuel Electron. La plupart de ces programmes seront d'ailleurs utilisables sans grande modification sur d'autres ordinateurs utilisant aussi le même BASIC, comme le BBC.

Les programmes proposés sont très courts, et l'ordre dans lequel ils sont présentés n'est bien évidemment pas arbitraire : il s'agit d'initier le lecteur, à travers la réalisation et l'utilisation de programmes de jeux de bonne qualité, à la connaissance et à la maîtrise progressive de cet outil extraordinaire qu'est un ordinateur individuel.

La brièveté des programmes proposés permettra de les taper facilement, sans risque d'erreur. Vous pourrez bien sûr les utiliser comme base pour la réalisation de jeux plus performants.

Les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver facilement les caractéristiques propres des instructions Basic qu'ils utilisent.

Les programmes n'utilisent pas toutes les particularités d'un appareil comme Electron, par ailleurs très bien détaillées dans le manuel de base.

Le niveau 5 donne une idée sur l'utilisation de certaines techniques un peu spécifiques, comme les caractères graphiques utilisateur et les définitions de procédures. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

Vous trouverez, à la fin de chaque texte de présentation, quelques suggestions pour des améliorations possibles, ou des modifications, qui pourraient rendre le jeu proposé plus attrayant, plus performant ou plus complet.

Organisation de l'ouvrage

Les jeux sont classés par ordre de difficulté croissante, suivant 5 niveaux :

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT, INPUT, IF...THEN, FOR...NEXT, GOTO, GOSUB, ON...GOTO, ON...GOSUB.**

- **Au niveau 2,** on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM, R(I), MID$.**

- **Au niveau 3,** on utilise la panoplie complète des instructions de traitement des chaînes de caractères : **MID$, RIGHT$, LEFT$, READ, DATA.**

- **Le niveau 4** fait appel à deux fonctions très particulières qui facilitent la réalisation de jeux interactifs : **INKEY$** et **GET$** et nous fait découvrir des instructions graphiques et sonores : **PLOT, SOUND.**

- Enfin, **le niveau 5** nous présente des instructions plus spécifiques : définition des caractères graphiques utilisateur, définition de procédures, gestion de touches utilisateur : **VDU, DEFPROC, * FX, DEFFN.**

Chaque jeu est présenté d'une façon très systématique, par une courte description, suivie de l'étude détaillée de chacune des lignes importantes du programme. Après quoi les indications nécessaires au bon fonctionnement du jeu sont données, ainsi que quelques éléments pouvant conduire à des extensions ou des améliorations du jeu proposé.

Le programme, fonctionnant sur Electron, est ensuite reproduit. Les listings ont été obtenus directement d'après les originaux de l'auteur sur une imprimante, ce qui devrait garantir contre tout risque de disfonctionnement. Les minuscules ne sont pas utilisées.

Tous les programmes (sauf mention particulière) fonctionnent en MODE 1 (quatre couleurs).

Niveau 1

Nom du programme	*Thèmes et particularités*
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre l'ordinateur.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, mois par mois.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours de 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège à l'aide de votre ordinateur.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque.
Pair - Impair	Un jeu de casino en trois temps (1 ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel pour un joueur.
Roulette	Une simulation élégante du jeu très connu.
Couloirs	Un jeu de logique, à découvrir.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.
Jackpot	Jeu de hasard, pour un joueur.

Instructions élémentaires

Nous ne retiendrons ici que les instructions fondamentales, ou instructions de programmation ; beaucoup d'autres commandes, ou fonctions particulières, seront utilisées à l'occasion, par exemple :

– ***RND pour obtenir un nombre aléatoire entre* Ø et 1.**

– ***SGN pour obtenir le signe d'une variable.***

– ***INT pour prendre la partie entière d'un résultat.***

– ***STRING$ pour former une chaîne composée de caractères identiques.***

– ***REM pour placer un commentaire.***

– ***END pour terminer un programme.***

– ***ASC pour obtenir le code ASCII d'un caractère.***

– ***CHR$ pour obtenir un caractère dont on connaît le code ASCII.***

Les instructions fondamentales de la programmation, utilisées dès le niveau 1, et présentes dans pratiquement tous les programmes sont :

LET (optionnelle) ; **PRINT ; INPUT ; GOTO ⎵ ; IF ⎵ THEN ⎵ ; FOR ... NEXT ; GOSUB ... RETURN.**

soit seulement sept instructions de base (et deux instructions dérivées) que nous allons maintenant examiner plus en détail.

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

C'est l'instruction par excellence des langages de programmation évoluée, puisque son rôle essentiel est de permettre, pour le programmeur, une gestion transparente de la mémoire. Cela signifie, en pratique, que pour ranger une quantité en mémoire, par exemple 12, il n'est pas nécessaire de savoir dans quelle case de la mémoire on va la déposer : il suffit de lui affecter un nom autorisé, et c'est par ce nom que la quantité sera par la suite retrouvée et manipulée.

– **Si la quantité est numérique,** le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

– **Si la quantité est alphanumérique,** la même règle s'applique pour le nom, qui devra cependant être suivi du symbole $, et la variable devra être écrite entre guillemets.

Exemple :
LET A1 = 12 ; la variable numérique A1 prend la valeur 12
LET NOM $ = "RENOIR" ; la variable alphanumérique NOM$ prend la valeur RENOIR.

Sur Electron, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc : A = 12 ; NOM$ = "TOTO" ; etc.

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

C'est une instruction de dialogue, qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul, ou une variable. C'est le seul moyen que possède l'ordinateur individuel pour faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant A1 = 12), PRINT A$ provoque l'affichage de BONJOUR (si l'on a fait A$ = "BONJOUR"). Utilisé après un effacement d'écran, le PRINT provoque l'affichage sur la 1re ligne, en haut, à gauche.

Un certain nombre de paramètres peuvent être utilisés pour obtenir sur l'écran un affichage convenable :

– **PRINT TAB C, L** provoque sur la ligne d'impression courante un affichage commençant à la cième colonne à partir de la gauche (Ø ⩽ C ⩽ 39) et à la lième ligne de l'écran (Ø ⩽ L ⩽ 31 en mode 1).

Ainsi, pour écrire le mot BONJOUR à la 5e colonne de la 8e ligne, on écrira : PRINT TAB (5,8) ; "BONJOUR"

– Un ; (**point-virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.

– La , (**virgule**) provoque une tabulation horizontale automatique prédéfinie.

Lors de la frappe, **l'instruction PRINT** pourra être abrégée en tapant **?** (**point d'interrogation**). Lors d'une impression, tout nombre est précédé d'un signe ; mais le signe positif n'est pas affiché un espace précédera donc automatiquement l'impression de tout résultat positif.

L'INSTRUCTION CLS

CLS efface l'écran et le colorie dans la couleur du fond.

L'INSTRUCTION INPUT OU INSTRUCTION D'ENTREE (CLAVIER)

Cette instruction, qui provoque un arrêt et une attente de l'appareil, ***permettra d'entrer un nombre, ou un texte, qui sera affecté à la variable suivant l'ordre INPUT.*** Ceci est fait en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque Electron rencontre une instruction INPUT, il s'arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l'aide de la touche **RETURN,** ce qui lui rendra la main.

Il est possible, comme pour le PRINT, d'afficher des messages avant l'entrée de la variable, et d'utiliser la fonction TAB.

On pourra, par exemple, écrire :

INPUT "VOTRE NOM, S'IL VOUS PLAIT" ; A$

Si le point-virgule est supprimé, Electron n'affichera pas de (?) point d'interrogation à la suite du texte.

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que ***chaque ligne de programme doit être précédée d'un numéro,*** ces lignes étant ensuite exécutées par l'ordinateur individuel en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors ***l'instruction GOTO, qui permet de détourner le fonctionnement du programme de son cours normal*** quitte d'ailleurs à le reprendre ensuite par un autre GOTO. Le renvoi doit, par ailleurs, impérativement se faire sur un numéro de ligne existant dans le programme (GOTO 2Ø, par exemple).

Pour donner un peu de souplesse à cette instruction de renvoi, il existe une instruction très puissante, qui permet de renvoyer à n'importe quel numéro de ligne, selon la valeur d'une variable numérique : **c'est l'instruction ON⌴ GOTO⌴ ;** on écrira par exemple :

ON X GOTO 1ØØ, 2ØØ, 3ØØ et le renvoi sera fait à la ligne 1ØØ si X vaut 1, à la ligne 2ØØ si X vaut 2, et 3ØØ si X vaut 3.

L'INSTRUCTION IF⌴ THEN⌴ OU INSTRUCTION DE TEST

On peut l'utiliser de deux façons : d'abord, en rupture de séquence conditionnelle, sous la forme :

IF ⌊condition = vraie⌋ THEN ⌊numéro de ligne⌋

C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de programme, si par exemple X = Ø, ou si A = B, etc. Les **opéra-**

teurs logiques NOT, OR ou **AND** peuvent d'ailleurs être utilisés dans l'écriture de la condition.

On pourra aussi utiliser le test sous la forme :

IF condition THEN instruction ;

l'instruction présentée à la suite de THEN peut être une affectation A = 12, une impression PRINT A, une entrée INPUT B $, ou même un autre test IF ⎵ THEN ⎵.

On peut par exemple écrire :

IF A > 3 THEN IF B < 7 THEN PRINT A + B

L'INSTRUCTION FOR...NEXT OU INSTRUCTION DE BOUCLES

On est très souvent conduit, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 6 à la case 2Ø, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une ***boucle,*** la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le ***pas*** de cette variable (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I = Ø TO 2Ø STEP 2
        Instructions
NEXT I
```

Les instructions contenues dans la boucle seront exécutées pour I = Ø, pour I = 2, ... et ceci jusqu'à I = 2Ø, soit 11 fois. La variable I peut être utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. Si l'on veut sortir de la boucle avant I = 2Ø par un test portant une quantité quelconque, il faut quand même passer par un **NEXT.**

Si ***le pas de la boucle (STEP)*** n'est pas précisé, il sera automatiquement pris égal à 1.

Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le NEXT.

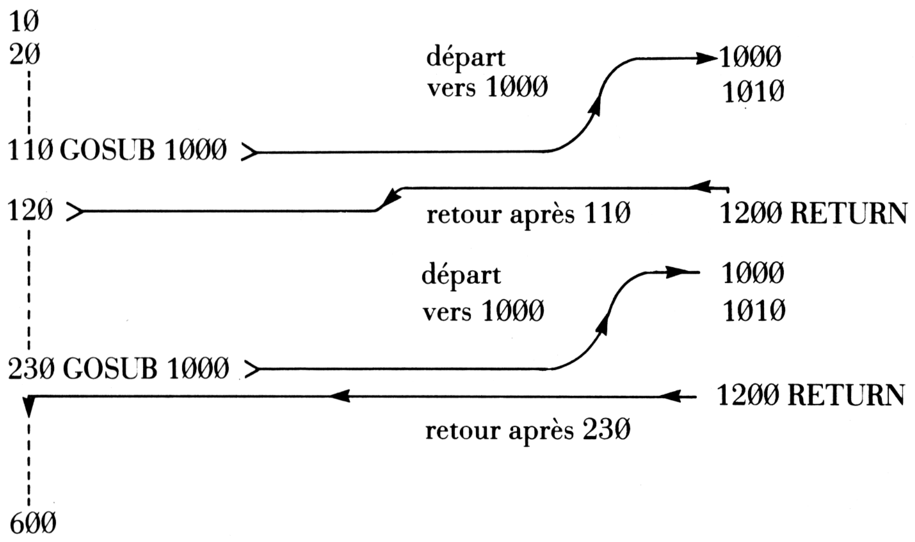
L'INSTRUCTION GOSUB OU INSTRUCTION D'APPEL DE SOUS-PROGRAMME

Lorsqu'une même suite d'instructions doit être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, on évitera de la réécrire plusieurs fois. On la rédige une fois pour toutes, et on la place en dehors du programme proprement dit (en lui donnant, par exemple, un numéro de ligne supérieur au numéro de la dernière ligne du programme). Indépendamment des difficultés que l'on pourra rencontrer au niveau du

nom des variables, lorsque l'on sera désireux d'utiliser cette séquence, il suffira de l'appeler par **l'instruction GOSUB numéro de ligne.** Le programme ira dès lors "faire un détour" dans la partie du programme appelé, mais il ne pourra revenir et poursuivre son déroulement normal que si il rencontre dans le programme appelé une instruction **RETURN,** ou instruction de retour.

Voici le schéma théorique de fonctionnement :

Programme principal



Fin du programme principal

1000

1200 RETURN

Sous-programme 1

Le sous-programme n'est écrit qu'une fois, mais appelé aussi souvent que l'on veut.

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite.

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction ON␣ GOSUB ␣.

Par exemple ON X GOSUB 1000, 2000, 3000 permettra d'exécuter le sous-programme placé en 1000 si X = 1, en 2000 si X = 2, en 3000 si X = 3.

Le nombre mystérieux

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction pour un joueur.
Difficulté du jeu : facile.

LE JEU :
On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).
On devra ensuite essayer de le découvrir, avec le moins d'essais possible, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

LE PROGRAMME :
La ligne 5Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 7Ø à 8Ø.
La ligne 9Ø teste si le résultat est exact.
La ligne 1ØØ signale un essai trop petit, et la ligne 11Ø un essai trop grand.
Le compteur d'essais est en ligne 12Ø.
Les lignes 2ØØ à 23Ø s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :
On lancera le programme par RUN.
On obtient un premier affichage NOMBRE DE CHIFFRES auquel on répond par un entier entre 1 et 7, suivi de RETURN.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.
On entre un nombre, et l'on presse RETURN.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste un affichage de VOUS AVEZ TROUVE... EN... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.
En fin de partie, on obtient la question AUTRE PARTIE (O/N) : on répond O pour faire une autre partie (suivi de RETURN).
On peut arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

```
   10 REM LE NOMBRE MYSTERIEUX
   20 CLS
   30 INPUT"NOMBRE DE CHIFFRES : ";N
   40 IFN>7THENRUN
   50 X=INT(RND(10^N-1))
   60 J=1
   70 PRINT"ESSAI N. ";J
   80 INPUT"VOTRE NOMBRE ... : ";M
   90 IF M=X THEN200
  100 IF M<X THENPRINT"===  TROP PETIT"
  110 IF M>X THENPRINT"===  TROP GRAND"
  120 J=J+1
  130 PRINT:PRINT
  140 GOTO70
  200 CLS
  210 PRINT:PRINT"VOUS AVEZ TROUVE ";X;"
 EN ";J;" ESSAIS."
  220 PRINT:INPUT"UNE AUTRE PARTIE ? (O/
N) : ";K$
  230 IFK$="O"THENRUN
>VDU12,3
```

Cube

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé se contente de réaliser à l'aide d'instructions PRINT une vue d'un cube en perspective cavalière.
Il pourrait être par exemple incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

LE PROGRAMME :
Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (3Ø-12Ø), et la boucle I (13Ø-18Ø).
Le nom de l'objet représenté est affiché en ligne 19Ø (ici : CUBE).

POUR JOUER :
Il suffit de lancer l'exécution par RUN, et de regarder.

EXTENSIONS POSSIBLES :
Le graphique permettra d'avoir un tracé plus précis.
On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```
   10 REM CUBE
   20 CLS
   30 FORI=0TO9
   40   PRINT TAB(7+I,0);"."
   50   PRINT TAB(I,7);"."
   60   PRINT TAB(7+I,10);"."
   70   PRINT TAB(I,17);"."
   80   PRINT TAB(0,7+I);"."
   90   PRINT TAB(10,7+I);"."
  100   PRINT TAB(7,I);"."
  110   PRINT TAB(17,I);"."
  120   NEXTI
  130 FORI=0TO7
  140   PRINT TAB(I,7-I);"."
  150   PRINT TAB(10+I,7-I);"."
  160   PRINT TAB(I,17-I);"."
  170   PRINT TAB(10+I,17-I);"."
  180   NEXTI
  190 PRINT TAB(25,18);"CUBE"
>VDU12,3
```

Chaud, froid

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un des plus vieux jeux du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications sous la forme TU BRULES, ou C'EST FROID, etc.
Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possible (votre score sera affiché en fin de partie).

LE PROGRAMME :
Les lignes 4Ø et 5Ø définissent la position de l'objet.
La ligne 9Ø calcule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 6Ø-7Ø.
La ligne 1ØØ regarde si vous avez gagné.
Les lignes 12Ø-13Ø normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 14Ø effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :
On lance le programme par l'ordre RUN, puis on introduit les dimensions de la grille (carrée) et l'on frappe RETURN.
Dès l'affichage suivant (DONNEZ X), on entre la coordonnée X supposée, suivie de RETURN.
On obtient ensuite l'affichage DONNEZ Y, et l'on entre Y suivi de RETURN.
On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de COMPLETEMENT GELE à BRULANT). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.
Pour sortir du programme avant la fin de la partie, il suffit de taper ESCAPE.

EXTENSIONS POSSIBLES :
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

```
   10 REM CHAUD,FROID
   20 CLS
   30 INPUT"DIMENSION DE LA GRILLE ... :
 "D
   40 X=INT(RND(D))
   50 Y=INT(RND(D))
   55 PRINTX;Y
   60 PRINT:INPUT"DONNEZ X : "A
   70 INPUT"DONNEZ Y : "B
   80 S=S+1
   90 T=ABS(X-A)+ABS(Y-B)
  100 IFT=0THEN6000
  110 PRINT"---  VOUS ETES ";
  120 N=INT(SQR(D)/T)+1
  130 IFN>5THENN=5
  140 IF T<=SQR(D) THEN ON N GOTO 1000,2
000,3000,4000,5000
  150 PRINT"COMPLETEMENT GELE"
  160 GOTO60
 1000 PRINT"A PEINE TIEDE"
 1010 GOTO60
 2000 PRINT"TIEDE"
 2010 GOTO60
 3000 PRINT"CHAUD"
 3010 GOTO60
 4000 PRINT"TRES CHAUD"
 4010 GOTO60
 5000 PRINT"BRULANT"
 5010 GOTO60
 6000 CLS
 6010 PRINT"VOUS AVEZ TROUVE ";X;",";Y;"
 EN ";S;" COUPS."
>VDU12,3
```

Les allumettes

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : contre Electron.
Difficulté du jeu : moyenne.

LE JEU :
Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence, ou s'il laisse commencer Electron. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé. Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :
Le jeu du joueur commence en ligne 7Ø.
Celui de l'Electron commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup de l'Electron est calculé dans une boucle non formelle, de la ligne 12Ø à la ligne 15Ø.
Le résultat est affiché par la ligne 1Ø2Ø.
Si la partie est terminée, la ligne 1Ø3Ø le détecte et envoie en ligne 2ØØØ, pour le message de fin.

POUR JOUER :
On commence l'exécution par RUN, et, à l'affichage du message NOMBRE TOTAL, on entre le nombre d'allumettes, suivi de RETURN. Puis on entre la prise maximum autorisée, suivie de RETURN. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis RETURN.
Après chaque coup, le nombre d'allumettes restantes est affiché, et Electron affiche VOUS RETIREZ lorsque c'est au joueur de prendre.
Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :
On pourra bien sûr envisager un affichage graphique couleur plus spectaculaire, imaginer de laisser le hasard choisir le nombre d'allumettes et la prise maximum, etc.

```
   10 REM LES ALLUMETTES
   20 CLS
   30 INPUT"NOMBRE TOTAL ....... : "N
   40 INPUT"PRISE MAXIMUM ...... : "I
   50 INPUT"QUI COMMENCE (VOUS:0 MOI:1)
: "D
   60 IF D=1 THEN110
   70 INPUT"VOUS RETIREZ ....... : "P
   80 IF P<>INT(P) OR P<1 OR P>N OR P>I
THEN 70
   90 D=ABS(D-1)
  100 GOTO1010
  110 S=I+D
  120 T=(N-S) / (I+D)
  130 IF T=INT(T) THEN200
  140 S=S-D
  150 GOTO120
  200 P=ABS(S-D)
  210 IF P=0 THENP=D
  220 D=0
 1000 PRINT"JE RETIRE : ";P
 1010 N=N-P
 1020 PRINT"IL RESTE  : ";N
 1030 IF N=0 THEN2000
 1040 GOTO60
 2000 CLS
 2010 IF D=1 THEN PRINT"===  J'AI GAGNE
..."
 2020 IF D<>1 THEN PRINT"===  BRAVO ..."
>VDU12,3
```

Allumettes à plusieurs

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (plusieurs joueurs).
Difficulté du jeu : assez difficile.

LE JEU :
Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer à Electron le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).

LE PROGRAMME :
Les lignes 10 à 40 permettent les différentes entrées.
Les lignes 70 à 100 réalisent les impressions.
La ligne 120 contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 180 gère les tours (remise à 1 du numéro du joueur).
La ligne 160 réalise le test de fin, et la ligne 210 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :
Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de RETURN, puis le nombre de joueurs (et RETURN).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par Electron.
Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR N° A GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES :
On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront ainsi éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
   10 REM ALLUMETTES A PLUSIEURS
   20 CLS
   30 INPUT"NB. TOTAL D'ALLUMETTES ... :
 "N
   40 INPUT"NOMBRE DE JOUEURS ........ :
 "J
   50 F=1
   60 K=1
   70 CLS
   80 PRINT:PRINT"TOUR DU JOUEUR N.";K
   90 PRINT"TOTAL RESTANT .......... : "
;N
  100 PRINT"MAXIMUM A ENLEVER ...... : "
;F
  110 INPUT"  TAPER VOTRE COUP : ";C
  120 IF C<1 OR C>F THEN110
  130 N=N-C
  140 IF N<0 THEN N=N+C:GOTO110
  150 F=C+C
  160 IF N=0 THEN200
  170 K=K+1
  180 IF K>J THENK=1
  190 GOTO70
  200 CLS
  210 PRINT"=== LE JOUEUR N.";K;" A GAGN
E ..."
>VDU12,3
```

Calendrier perpétuel

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : gadget publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

LE PROGRAMME :
Les lignes 3Ø et 4Ø vous permettront d'entrer les mois et l'année.
Les lignes 5Ø, 6Ø et 7Ø calculent le numéro du premier jour du mois.
La ligne 8Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (12Ø-15Ø) réalise l'impression du calendrier proprement dit.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1983, suivie de RETURN.
On obtient alors la question MOIS, à laquelle il convient de répondre par le numéro du mois, suivant l'ordre usuel (1 pour JANVIER, 2 pour FEVRIER, etc.).
On fait ensuite RETURN, et le calendrier est alors affiché sur l'écran.

EXTENSIONS POSSIBLES :
On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

```
   10 REM CALENDRIER PERPETUEL
   20 CLS
   30 INPUT"TAPER L'ANNEE .... : "A
   40 INPUT"TAPER LE MOIS .... : "M
   50 D=INT(365.25*(A+(M<3)))+1
   60 D=D+INT(30.6*(M+1-(M<3)*12))-INT((
INT((A+(M<3))/100)-7)*.75)
   70 D=D MOD 7
   80 M=31+(M=4 OR M=6 OR M=9 OR M=11) +
 (M=2)*3
   90 PRINT
  100 PRINT"DIM LUN MAR MER JEU VEN SAM"
  110 PRINT
  120 FORA=1TOM
  130   PRINTTAB(4*(D+A-1));A;
  140   IF D+A-1=6 THENPRINT:D=D-7
  150   NEXTA
  160 PRINT
>VDU12,3
```

Bataille navale

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 1Ø cases sur 1Ø cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et Electron répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :
La boucle I (4Ø-7Ø) dessine le terrain de jeu.
Les lignes 9Ø-1ØØ définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 11Ø et 12Ø, et la ligne 13Ø affiche le coup sur l'écran. La distance est calculée en ligne 14Ø, et la ligne 15Ø regarde si l'on a ou non fait mouche.
La ligne 17Ø intervient pour déplacer le navire si les coups sont rapprochés.
La ligne 18Ø permet l'affichage de la distance.

POUR JOUER :
Le terrain de jeu est affiché après avoir frappé RUN. A l'affichage de LIGNE, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer. On tape ensuite RETURN et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de RETURN.
Si on a tiré trop près, on obtient l'affichage PFFFF... qui signifie que le navire ennemi a changé d'emplacement.
Si on a tiré sur la bonne case, on obtient l'affichage du nombre total de tirs effectués pour gagner.

EXTENSIONS POSSIBLES :
Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre Electron, mettre plusieurs navires, des sous-marins, etc.

```
10 REM BATAILLE NAVALE
20 CLS
30 PRINT"   0  1  2  3  4  5  6  7  8
9"
40 FORI=0TO9
50   PRINT;I;"  .  .  .  .  .  .  .
.  .  ."
60   PRINT
70   NEXT
80 S=1
90 X=INT(RND(10)-1)
100 Y=INT(RND(10)-1)
105 PRINTTAB(0,20)X Y
110 INPUT TAB(0,21);"LIGNE ...... : "H
120 INPUT TAB(20,21);"COLONNE .... : "
V
130 PRINTTAB(3*V+3,2*H+1);S
140 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V))
)
150 IFD=0THEN200
160 S=S+1
170 IFD=1THENPRINTTAB(0,22);"PFFFF...
                    ":GOTO90
180 PRINTTAB(0,22);"DISTANCE      : ";D
190 GOTO110
200 CLS
210 PRINT"===  TROUVE EN ";S;" COUPS."
)VDU12,3
```

Golf

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.
Mais attention, il peut y avoir un certain aléa sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe −).

LE PROGRAMME :
La ligne 7Ø permet de définir la position du trou, et la ligne 8Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 12Ø.
La boucle I (14Ø-19Ø) réalise le déplacement de la balle.
La ligne 2ØØ regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 25Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (4Ø-25Ø) ; fin de boucle en 27Ø, avec affichage de fin).

POUR JOUER :
On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 3ØØ...).
Dès que l'on aura tapé RETURN, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.
Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.

EXTENSIONS POSSIBLES :
Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcade), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

```
   10 REM GOLF
   40 FORS=1TO18
   50    CLS
   60    X=1
   70    T=INT(RND(29)+8)
   80    PRINTTAB(0,6)"==================
======================"
   90    PRINTTAB(T,6);" ":PRINTTAB(X,5);
"O "
  100    PRINTTAB(15,8);"            "
  110    PRINTTAB(05,8);"TROU ";S;:INPUT"
 FORCE "R
  120    F=INT(R*.16-(ABS(R)*(SQR(RND(1))
+.1)^2)/1000)
  130    IF F=0 THEN100
  140    FOR I = SGN(F) TO F STEP SGN(F)
  150      X=X+SGN(F)
  160      IF X<39 AND X>=1 THEN PRINTTAB
(X-SGN(F),5);" ":PRINTTAB(X,5);"O"
  170      IF X>=38 THEN X=38
  180      IF X<=1 THENX=1
  190      NEXTI
  200    IF X=T THEN230
  210    E=E+1
  220    GOTO80
  230    PRINTTAB(X,5);" ":PRINTTAB(X,6);
"O"
  240    FORP=1TO500:NEXTP
  250    NEXTS
  260 CLS
  270 PRINT:PRINT"18 TROUS EN ";E;" COUP
S."
>VDU12,3
```

Solfège

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée ; après l'affichage de la note, vous devez entrer son nom en clair et votre score sera affiché en permanence.
Le programme est initialement prévu pour la clé de sol, mais il sera très facilement adaptable à toute autre clé.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine une portée.
La note est choisie en ligne 8Ø et affichée en lignes 9Ø-1ØØ-11Ø.
La ligne 12Ø affiche le score.
La ligne 13Ø affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 14Ø s'occupe de la fin de la partie.
Les lignes 15Ø et 16Ø effectuent le décodage de la note entrée.
La ligne 19Ø contrôle le résultat.

POUR JOUER :
Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.
Vous répondrez à la question NOTE en donnant le nom en clair de la note (do, ré, mi, fa, sol, la ou si), suivi de RETURN.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre Ø suivi de RETURN, à la question NOTE.

EXTENSIONS POSSIBLES :
On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés. On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple). On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

```
  10 REMSOLFEGE
  30 S=0:P=0
  40 CLS
  50 FORI=1TO5
  60   PRINTTAB(0,2*I+1)"------------
--------------------------"
  70   NEXTI
  80 X=INT(RND(12)-1)
  90 PRINT TAB(18,X)"!"
 100 PRINT TAB(18,X+1)"!"
 110 PRINT TAB(17,X+2)"O"
 120 PRINT TAB(0,21)"SCORE ";S;" SUR ";
P;" ESSAIS."
 130 INPUT"CLE DE SOL. QUELLE EST LA NO
TE : "R$
 140 IFR$="0"THENEND
 150 Y=(1 AND R$="FA") + (2 AND R$="MI"
) + (3 AND R$="RE") + (4 AND R$="DO")
 160 Y=Y + (5 AND R$="SI") + (6 AND R$=
"LA")
 170 P=P+1
 180 IF R$="SOL" THEN R$="SO"
 190  IF Y=X MOD 7 THEN S=S+1
 200 GOTO 40
>VDU12,3
```

Le jeu du 21

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu contre Electron.
Difficulté du jeu : facile.

LE JEU :
On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer Electron qui essaiera alors d'avoir un meilleur score.
A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

LE PROGRAMME :
Les lignes 8Ø et 9Ø réalisent le contrôle de celui qui doit jouer (joueur ou Electron).
Le sous-programme (1ØØØ-1Ø1Ø) lance les dés.
Les lignes 11Ø-12Ø calculent le total fait par le joueur et l'affichent.
Le total de l'Electron est calculé en ligne 19Ø et affiché par la ligne 2ØØ.
La ligne 22Ø contrôle la fin du jeu pour Electron.
Les lignes 26Ø-27Ø affichent les résultats, et la ligne 3ØØ cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 32Ø, et le total par la ligne 35Ø.
La ligne 38Ø contrôle la fin de partie.

POUR JOUER :
Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de RETURN, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de RETURN, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et RETURN pour laisser jouer Electron (qui tient ici le rôle de la banque).
Le gain (ou la perte, s'il y a le signe –) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question ON CONTINUE (O/N).

EXTENSIONS POSSIBLES :
Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence. On pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de Black Jack.

```
   10 REM JEU DU 21
   40 G=0
   50 CLS
   60 T=0
   70 INPUT"--- VOTRE MISE : "M
   80 INPUT"    TAPER 1 POUR TIRER, 0 PO
UR FINIR "R
   90 IF R=0 THEN 150
  100 GOSUB 1000
  110 T=T+X
  120 PRINT "    TOTAL = ";T
  130 GOTO 80
  150  CLS
  160 PRINT"====  LA BANQUE JOUE ..."
  170 S=0
  180 GOSUB1000
  190 S=S+X
  200 PRINT"TOTAL DE LA BANQUE :";S
  210 FOR I=1TO 800: NEXT I
  220 IF S>T OR S>17 THEN250
  230 GOTO180
  250 CLS
  260 PRINT"== VOUS AVEZ ";T
  270 PRINT
  280 PRINT"== LA BANQUE A ";S
  290 PRINT
  300 IF S>21 OR (S<T AND T<=21) THEN 32
0
  310 M=-M
  320 PRINT"==--- GAIN : ";M
  330 PRINT
  340 G=G+M
  350 PRINT"VOTRE TOTAL ACTUEL ..... : "
;G
  360 PRINT
  370 INPUT"ON CONTINUE ? (O/N) : "R$
  380 IF R$="N" THEN END
  390 GOTO50
 1000  X=RND(6)
 1010 RETURN
>VDU12,3
```

Pair-impair

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, pour un ou plusieurs joueurs. Electron lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :
Les dés sont tirés par un sous-programme placé aux lignes 1000 et 1010.
La ligne 80 tient le compte du nombre de parties, et les lignes 180, 260 et 290 comptabilisent les points marqués.
La parité du total est déterminée par les lignes 130 et 140 (la ligne 140 compare avec la réponse du joueur).
La ligne 400 et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 440 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :
On lance le programme par l'instruction RUN
Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant RETURN après chaque réponse.
Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, 6 SUR 10.
On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de RETURN pour arrêter.

EXTENSIONS POSSIBLES :
Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs.
On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

```
   10 REM PAIR-IMPAIR
   40 T=0:R=0
   50 GOSUB1000
   60 A=X
   70 GOSUB1000
   80 R=R+6
   90 S=A+X
  100 CLS
  105 PRINTA,X
  110 INPUT"TOTAL PAIR (0) OU IMPAIR (1)
 : "N
  120 PRINT
  130 P= S MOD 2
  140 IF N=P THEN 170
  150 PRINT"===  FAUX"
  160 GOTO 190
  170 PRINT"===  EXACT"
  180 T=T+1
  190 PRINT
  200 INPUT"TOTAL PRESUME : "N
  210  IF N=S THEN 250
  220 PRINT"===  PERDU"
  230 FORI=1TO1000 : NEXTI
  240 GOTO 400
  250 PRINT
  260 T=T+2
  270 PRINT:INPUT"BRAVO ! DONNEZ UN DE
 "N
  280 IF N<>A AND N<>S-A THEN 220
  290 T=T+3
  300 PRINT
  310 PRINT"===  BRAVO":GOTO230
  320 PRINT"==--- GAIN : ";M
  330 PRINT
  340 G=G+M
  350 PRINT"VOTRE TOTAL ACTUEL .....
;G
  360 PRINT
  370 INPUT"ON CONTINUE ? (O/N) : "R$
  380 IF R$="N" THEN END
  390 GOTO50
  400 CLS
  410 PRINT"VOUS AVEZ UN TOTAL DE ";T;"
SUR ";R;"."
  420 PRINT
  430 INPUT"ON CONTINUE ? (O/N) : "R$
  440 IFR$="N"THEN END
  450 GOTO 50
 1000  X=RND(6)
 1010 RETURN
>
```

Devine un chiffre

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît deux fois il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :
Le tirage des dés est réalisé par un sous-programme placé en lignes 1000 et 1010.
L'affichage est réalisé par les lignes 150 et 160.
Les lignes 180 et 210 calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 230 pour les gains, et en ligne 510 pour les pertes.
La variable R$, testée en ligne 550, détermine la poursuite éventuelle du jeu.

POUR JOUER :
Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ; on répond par un nombre suivi de RETURN. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de RETURN ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte. Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O/N) ; taper O pour continuer, N pour arrêter (suivi de RETURN).

EXTENSIONS POSSIBLES :
On pourra facilement introduire des contrôles (mise minimum, coup proposé de 1 à 6, etc.). On pourra aussi dessiner des dés avec un graphisme plus élaboré, introduire la possibilité de jouer à plusieurs, faire même participer Electron.

```
   10 REM DEVINE UN CHIFFRE
   40 G=0
   50 CLS
   60 INPUT"VOTRE MISE ..... : "N
   70 PRINT
   80 INPUT"DONNEZ UN CHIFFRE DE 1 A 6 :
 "D
   90 GOSUB1000
  100 A=X
  110 GOSUB1000
  120 B=X
  130 GOSUB1000
  140 C=X
  150 PRINT:PRINT:PRINT"          A
   B          C"
  160 PRINT:PRINTA,B,C
  170 T=0
  180 IF D=A THEN T=T+N
  190 IF D=B THEN T=T+N
  200 IF D=C THEN T=T+N
  210 IF T=0 THEN 500
  220 PRINT:PRINT:PRINT"===  VOUS GAGNEZ
 $";T
  230 G=G+T
  240 GOTO 520
  500  PRINT:PRINT:PRINT"===  PERDU ..."
  510 G=G-N
  520  PRINT:PRINT:PRINT"VOTRE TOTAL ACT
UEL EST DE $";G
  530 PRINT
  540 INPUT"ON CONTINUE ? (O/N) : "R$
  550  IFR$="N"THEN END
  560  GOTO 50
 1000  X=RND(6)
 1010 RETURN
>VDU12,3
```

Le jeu de Wytthof

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
On dispose d'une grille carrée de 1Ø cases sur 1Ø cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale, bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :
Les lignes 4Ø, 5Ø et 6Ø permettent de définir une position de départ, en dehors de la diagonale.
La ligne 8Ø et la boucle I (9Ø-11Ø) dessinent le terrain de jeu.
La ligne 115 affiche l'indication des trois directions possibles par appel du sous-programme 2ØØ-25Ø.
La ligne 13Ø imprime la position du point mobile.
Le test de fin est effectué en ligne 14Ø.
Le changement de joueur est assuré par la ligne 18Ø.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1ØØØ, 2ØØØ et 3ØØØ.

POUR JOUER :
Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR 1 et en dessous la question DIRECTION (1, 2, 3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de RETURN.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9, 9).

EXTENSIONS POSSIBLES :
On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait à Electron de jouer, etc.

```
   10 REM LE JEU DE WYTTHOF
   40 X=RND(5)-1
   50 Y=RND(5)-1
   60 IF X=Y THEN 50
   70 CLS
   80 PRINT"     0  1  2  3  4  5  6  7
 8  9"
   90 FORI=0TO9
  100   PRINT:PRINT;I;"    .  .  .  .  .
 .  .  .  .  ."
  110   NEXTI
  115 GOSUB 200
  120 H=1
  130 PRINT TAB(3*X+5,2*Y+2);"+"
  140 IF X=9 AND Y=9 THEN END
  150 PRINT TAB(0,22);"JOUEUR N.";H;:INP
UT" - DIRECTION (1,2,3) : "D$
  160 D=VAL(D$)
  170 ON D GOSUB 1000,2000,3000
  180 H=2-ABS(H-1)
  190 GOTO130
  200 PRINT TAB(35,4);" ---1"
  210 PRINT TAB(35,5);"!\"
  220 PRINT TAB(35,6);"! \"
  230 PRINT TAB(35,7);"!  \"
  240 PRINT TAB(35,8);"3   2"
  250 RETURN
 1000 X=X+1
 1010 RETURN
 2000 GOSUB1000
 3000 Y=Y+1
 3010 RETURN
>VDU12,3
```

Rebonds

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse.
Difficulté du jeu : moyenne.

LE JEU :
Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

LE PROGRAMME :
La ligne 5Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 6Ø et 7Ø définissent les positions initiales de la balle et de la cible.
Les lignes 8Ø et 9Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 1ØØ.
La boucle J (13Ø-17Ø) réalise le déplacement de la balle.
La ligne 16Ø imprime la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 15Ø, par renvoi dans une partie du programme indépendante, située aux lignes 22Ø à 24Ø.
La ligne 18Ø effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 23Ø.

POUR JOUER :
On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à Z), et on l'entre suivi de RETURN.
La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.
On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
Il sera très intéressant, en utilisant un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille. On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
   10 REM REBONDS
   40 CLS
   50 PRINT" 123456789ABCDEFGHIJKLMNOPQR
STUVWXYZ"
   60 X=RND(18)+1
   70 Y=RND(18)+1
   80 PRINT TAB(0,X);"#"
   90 PRINT TAB(39,Y);"+"
  100 INPUT TAB(0,23);"VOTRE COUP ... :
"R$
  110 S=0
  120 T=X
  130 FORJ=1TO39
  140   T=T-X/(ASC(R$)-ASC("0")+7*(R$)="
A"))
  150   IF T)=22 OR T(0 THEN 220
  160   PRINT TAB(J,T);"#"
  170   NEXTJ
  180 IF ABS(T-Y)(3 THEN PRINT"== BRAVO
: ";S;" REBONDS"
  190  INPUT"UNE AUTRE PARTIE ? (O/N) :
"R$
  200 IF R$="N" THEN END
  210 RUN
  220 X=-X
  230 S=S+1
  240 GOTO170
)VDU12,3
```

Roulette

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Une boule se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36 ainsi que trois cases supplémentaires le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour, on peut ensuite la relancer pour une autre partie.

LE PROGRAMME :
La boule K (5Ø-11Ø) dessine le terrain de jeu, auquel la ligne 12Ø rajoute les trois symboles P, ØØ et I.
La boucle K (13Ø-2ØØ) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 16Ø et 18Ø.
Le sous-programme 3ØØ-31Ø réalise une légère temporisation.
La ligne 21Ø tire la position d'arrêt.
La ligne 23Ø gère les essais suivants.

POUR JOUER :
Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf "N") pour la relancer. Taper N pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, introduire différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

```
   10 REMROULETTE
   20 P=0
   40 CLS
   50 FOR K=1TO4
   60   FORI=1TO9
   70     S=I+I
   80     PRINTTAB(4*I);(S AND K=1)+(S+1
8 AND K=2)+(S-1 AND K=3)+(S+17 AND K=4);
   90     NEXTI
  100   PRINT:PRINT:PRINT:PRINT
  110   NEXTK
  120 PRINT TAB(0,2);"P":PRINTTAB(0,6);"
00":PRINTTAB(0,10);"I"
  130 FORK=1TO4
  140   FORI=1TO10
  145     IF I=10 AND K<4 THEN PRINT TAB
(0,4*K-1);:GOTO160
  150     PRINTTAB(4*I,4*K-3);
  160     X=X-1:SOUND1,1,100,1
  170     GOSUB300
  180     IF X=1 THEN PRINT"*":GOTO230
  190     NEXTI
  200   NEXTK
  210 X=RND(38)-1
  220 GOTO120
  230 INPUTTAB(0,20);"AUTRE ESSAI ? (O/N
) : "R$
  240 IF R$="N" THEN END
  250 GOTO20
  300 FORJ=1TO 50+2*P:NEXTJ: P=P+1
  310 RETURN
>VDU12,3
```

Couloirs

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche. De plus, un tourniquet analogue a été disposé dans le couloir central, pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?

LE PROGRAMME :
Les lignes 3Ø à 7Ø dessinent les couloirs (et les trois tourniquets).
La ligne 1ØØ attend votre choix.
Le décodage du chemin est fait par Electron en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 13Ø à 27Ø).

POUR JOUER :
Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question de l'Electron.
Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre Electron.

EXTENSIONS POSSIBLES :
Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées et d'autres sorties, et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur. On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.).

```
   10 REM COULOIRS
   20 CLS
   30 PRINT"----------------------"
   40  PRINT"B        ----------  X  2"
   50 PRINT"--------- X      -----"
   60 PRINT"A        ----------  X  1"
   70 PRINT"----------------------"
   80 I=0:J=0:K=0
   90 R$="":T$=""
  100 INPUTTAB(0,13);"ENTREE ? (1 OU 2)
:"C$
  110 IF C$<"1" OR C$>"2" OR LEN(C$)<>1
THEN 100
  120 R$=R$+C$
  130 PRINT:PRINTR$
  140 IF C$="1" THEN 190
  150 J=NOT(J)
  160 IF J<>0 THEN 250
  170 S$="B"
  180 GOTO220
  190 I= NOT(I)
  200 IF I<>0 THEN 250
  210 S$="A"
  220 T$=T$+S$
  230 PRINT TAB(0,17);"SORTIE EN ";S$:PR
INT:PRINTT$
  240 GOTO 100
  250 K=NOT(K)
  260 IF K<>0 THEN 170
  270 GOTO210
>VDU12,3
```

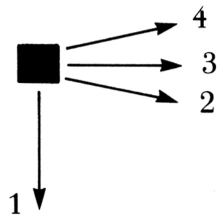
Le jeu de la fève

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (Ø, Ø), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale haut et droite.
On joue en principe avec la règle : le premier arrivé en (9, 9) a perdu.

LE PROGRAMME :
La ligne 3Ø et la boucle I (4Ø-6Ø) dessinent le terrain de jeu.
La ligne 1ØØ réalise le test de fin de partie.
La ligne 13Ø permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 14Ø et 15Ø assurent le changement de joueur.

POUR JOUER :
On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIRECTION (1, 2, 3, 4) ; les 4 directions proposées correspondent aux déplacements suivants :



Le joueur 1 entre son déplacement, et tape RETURN.
Le mouvement correspondant est réalisé, puis on obtient l'affichage JOUEUR 2 et DIRECTION.
Le joueur 2 procède de même, et ainsi de suite, jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.

EXTENSIONS POSSIBLES :
On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).
On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.

```
   10 REM LE JEU DE LA FEVE
   20 CLS
   30 PRINT"    0  1  2  3  4  5  6  7
8  9"
   40 FORI=1TO9
   50   PRINT:PRINT;I;"   .  .  .  .  .
 .  .  .  .  ."
   60   NEXTI
   65 PRINT TAB(0,23);"JOUEUR .  DIRECTI
ON *-3":PRINTTAB(21,22);"/":PRINT TAB(22
,21);"4":PRINT TAB(20,24);"|\":PRINTTAB(
20,25);"1 2"
   70 X=4:Y=1
   80 J=1
   90 PRINTTAB(X,2*Y);"#"
  100 IF X=31 AND Y=9 THEN 200
  110 PRINT TAB(7,23);J: INPUTTAB(25,23)
;".... "I$
  120 I=VAL(I$)
  130 ON I GOSUB 1000,2000,3000,4000
  140 J=J+1
  150 IF J=3 THEN 80
  160 GOTO 90
  200 PRINTTAB(0,30);"=== LE JOUEUR ";J;
" A GAGNE ..."
  210 END
 1000 Y=Y+1
 1010 RETURN
 2000 GOSUB1000
 3000 X=X+3
 3010 RETURN
 4000 GOSUB3000
 4010 Y=Y-1
 4020 RETURN
>VDU12,3
```

Jeu de l'oie aléatoire

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".
La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par Electron ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME :
La ligne 6Ø permet d'appeler le sous-programme 1ØØØ à 11ØØ qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 7Ø s'occupe du changement de joueur, et la ligne 8Ø effectue le lancer de dé.
La ligne 9Ø traite le cas où l'un des joueurs termine.
La ligne 13Ø tire un nombre aléatoire entre Ø et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 1ØØØ et 1Ø1Ø font avancer les pions, et les lignes 1Ø2Ø à 1Ø8Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :
On lancera le jeu par RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2 :...
Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...
Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.
On peut aussi s'arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut agrandir le terrain de jeu, introduire davantage d'aléas, changer leur nature. On pourrait aussi choisir les cases "chance" en début de jeu, ajouter des affichages en conséquence, etc.

```
   10 REMJEU DE L'OIE ALEATOIRE
   40 CLS
   50 X=0:Y=0:J=1:I=X
   60 GOSUB1000
   70 J=ABS(J-1)
   80 I=RND(6)
   90  IF X=35 OR Y=35 THENPRINTTAB(0,22
);CHR$(7);"=== BRAVO":END
  100  PRINTTAB(0,12);"JOUEUR ";J+1;" --
- VOTRE COUP : ";I
  110 INPUTTAB(0,14);"APPUYER SUR 'RETUR
N' : "R$
  120 GOSUB1000
  130 I=RND(4)-1
  140 F=RND(5)
  150 ON F GOSUB 200,300,400,500,160
  160 GOTO60
  200 I=I+3
  210 RETURN
  300 I=-I
  310 RETURN
  400 I=I*2
  410 RETURN
  500  IF J=0 THEN X=RND(30)-1:RETURN
  510 Y=RND(30)-1:RETURN
 1000 IF J=0 THEN X=X-I*(X+I<=35)
 1010 Y=Y-J*I*(Y+I<=35)
 1020 PRINTTAB(0,3);"0123456789ABCDEFGHI
JKLMNOPQRSTUVWXYZ"
 1030 FORI=0TO35
 1040   COLOUR 3:PRINTTAB(I,5);"-";
 1050   IF I=X THEN COLOUR 1:PRINTTAB(I,
5);"1";
 1060   COLOUR 3:PRINTTAB(I,6);"-";
 1070   IF I=Y THEN COLOUR 2:PRINTTAB(I,
6);"2";
 1080   NEXTI
 1090 FORI=1TO600:NEXT
 1100 COLOUR 3
 1110 RETURN
>VDU12,3
```

Jackpot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (un joueur).
Difficulté du jeu : facile.

LE JEU :
C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue).
Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :
Les instructions 5Ø à 11Ø permettent le dessin de la machine.
La mise sera entrée en ligne 12Ø, et testée en ligne 13Ø.
Les lignes 14Ø, 16Ø et 18Ø permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 45Ø.
Le résultat est affiché par la ligne 31Ø, puis testé en ligne 32Ø, et en ligne 42Ø, si l'on a gagné.

POUR JOUER :
Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape RETURN, et on entre une première mise (différente de Ø), suivie de RETURN.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau, faire RETURN et ainsi de suite.
Pour terminer, il suffit de répondre F à la question CONTINUER.
Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable Jackpot.

```
   10 REM JACKPOT
   20 CLS
   30 INPUTTAB(0,5);"POT DE DEBUT : $"P
   40 CLS
   50   PRINTTAB(10);" ____________   ^
"
   60   PRINTTAB(10);"! __________ !  #
"
   70  PRINTTAB(10);"!!   !   !   !!  #"
   80   PRINTTAB(10);"!! . ! . ! . !!==#
"
   90  PRINTTAB(10);"!!___!___!___!!"
  100  PRINTTAB(10);"!_____________!"
  110   PRINTTAB(10);"!############!"
  120 INPUTTAB(0,10);"VOTRE MISE : $"M
  130 IF M<=0 OR M>P THEN 120
  140 GOSUB450
  150 A=X
  160 GOSUB450
  170 B=X
  180 GOSUB450
  190 PRINTTAB(27,0);" ":PRINTTAB(27,1);
"^"
  200 GOSUB470
  210 PRINTTAB(27,1);" ":PRINTTAB(27,2);
"^"
  220 GOSUB470
  230 PRINTTAB(27,2);" ":PRINTTAB(27,3);
"^"
  240 GOSUB470
  250  PRINTTAB(27,2);"^":PRINTTAB(27,3)
;"#"
  260 GOSUB470
  270 PRINTTAB(27,1);"^":PRINTTAB(27,2);
"#"
  280 GOSUB470
  290 PRINTTAB(27,0);"^":PRINTTAB(27,1);
"#"
  300 GOSUB470
  310 PRINTTAB(13,3);A;TAB(17,3);B;TAB(2
1,3);X
  320 IF A=B AND A=X THEN 410
  330 N=-M
  340 P=P+N:PRINTTAB(0,12);"---  VOTRE G
AIN ... : $";N;"    "
  350 PRINTTAB(0,14);"---  IL VOUS RESTE
 $";P;"    "
  360 IF P=0 THEN 400
  370 PRINT TAB(0,17);"TAPER 'RETURN' PO
UR CONTINUER :"
```

```
  380 INPUTTAB(0,18);"                    'F' POU
R TERMINER  :"R$
  390 IF R$<>"F" THEN 120
  400 END
  410 N=M*40
  420 IF A=7 THEN N=M*500
  430 GOTO340
  440 END
  450 X=RND(10)-1
  460 RETURN
  470 SOUND1,1,100,2
  480 FOR K=1 TO 100
  490   NEXT K
  500 RETURN
>VDU12,3
```

Niveau 2

Nom du programme	*Thèmes et particularités*
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Visions	Un jeu sur les mots, pour l'ordinateur.
421	Une simulation simplifiée, plusieurs joueurs.
L'électricien fou	Logique et difficile, pour un seul joueur.
La course au 20	Jeu de réflexion, pour deux joueurs ou plus.
Musique	Révisez vos notions de solfège, grâce à l'ordinateur.
Poker	Une base de départ, pour mettre au point votre programme.
Le jeu de saute-mouton	Un jeu de réflexion, pour deux joueurs.
Lettres	Un jeu sur les mots, assez simple.
Espion	Dites-lui votre amour en code.
Accrochez les wagons	Pour les plus jeunes ; très visuel.
Tirage d'un dé truqué	Révisez vos connaissances en statistiques.
Tour de cartes	Votre ordinateur a plus d'un tour dans son sac.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
L'indésirable	Vous êtes propriétaire. Arriverez-vous à mettre la main sur ce locataire indésirable ?
Compose	Faire de la musique, c'est d'abord l'écrire...
Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
Le jeu de la vie	Un grand classique, en deux versions : manuelle et automatique.

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne

L'INSTRUCTION DIM OU INSTRUCTION DE DIMENSIONNEMENT D'UN TABLEAU

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par Electron que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (3Ø, 2) (ou DIM RD (2, 3Ø)). Ainsi, avec DIM RD (3Ø, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, avec 255 caractères au maximum.

Ainsi DIM NOM$ (12) réserve 12 emplacements en mémoire, pour des mots de 255 caractères au maximum.

ATTENTION ; Sur Electron, il est obligatoire de dimensionner un tableau, même s'il contient moins de 1Ø éléments.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (I)

Si l'on a un tableau à une dimension, numérique ou alphanumérique, défini par une instruction comme DIM R (3Ø) ou DIM N$ (2Ø), le Ième élément de ce tableau est repéré par R (I) (ou N$ (I)) ; il sera alors manipulé exactement comme une variable ordinaire (affectation, calcul, tests, etc.).

Ainsi INPUT R (3) permet d'entrer à la main la valeur du 3[e] élément numérique du tableau R. PRINT R$ (5) permet l'affichage du 5[e] élément du tableau alphanumérique R$, etc.

LA RECHERCHE D'UN ELEMENT DANS UNE CHAINE : MID$

Pour extraire le Ième élément de la chaîne alphanumérique A$, on utilisera l'instruction MID$.

Ainsi, si A$ = "BONJOUR", MID$ (A$, 3, 1) donne le caractère N ; de plus MID$ (A$, 3, 3) donne la sous-chaîne "NJO"et MID$ (A$, 6) la sous-chaîne "UR".

Dé truqué

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre Ø et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

LE PROGRAMME :
Il utilise le générateur aléatoire de l'Electron supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (7Ø-14Ø) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 11Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 9Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (18Ø-2ØØ), la ligne permettant le truquage se trouvant en 19Ø. Un nouveau tirage pourra être effectué sur pression de RETURN (ligne 22Ø).

POUR JOUER :
Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre Ø et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.
Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé. Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser la touche RETURN pour effectuer les tirages suivants. On arrête le programme en tapant ESCAPE.

EXTENSIONS POSSIBLES :
Il sera évidemment bien tentant d'utiliser sournoisement ce petit programme pour améliorer vos chances de gain dans un jeu de hasard utilisant des lancers de dés. On pourra alors envisager de figer les données définissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondant à une série de lancers de dés sous une forme spectaculaire, qui mette le truquage en évidence...

```
   10 REM DE TRUQUE
   30 DIM E(6)
   40 CLS
   50 P=0
   70 FOR I=1 TO 6
   80   PRINTTAB(17,3);"
 "
   90    PRINTTAB(0,3);"PROBABILITE DU D
E ";I;:INPUT" : "E(I)
  100   P=P+E(I)
  110   IF E(I)<0 OR E(I)>1 OR P>1 THEN
240
  120   PRINT TAB(6*I-6,6);"DE ";I
  130   PRINT TAB(6*I-6,8);E(I)
  140   NEXTI
  150 IF P<.9 THEN PRINTCHR$(7);:GOTO40
  160 FORI=1TO500 : NEXT I
  170 N=RND(6)
  180 FORI=1 TO 6
  190   IF N=I AND RND(1)>E(I) THEN I=6:
NEXT I:GOTO170
  200 NEXT I
  210 PRINTTAB(0,15);"TIRAGE : ";N;"    "
  220 PRINT:INPUT"TAPER 'RETURN' POUR UN
 AUTRE TIRAGE "R$
  230 GOTO 170
  240 PRINTTAB(0,4)"=== IMPOSSIBLE : 0<=
P<=1"
  250 P=P-E(I)
  260 FORJ=1TO1500:NEXT J
  270 PRINTTAB(0,4)"
    "
  280 GOTO90
>VDU12,3
```

Visions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Electron vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :
Une boucle I (4Ø-12Ø) se charge de réaliser les dessins.
La ligne 5Ø permet de transformer en nombre chaque lettre du mot entré au début (ligne 3Ø). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 6Ø à 11Ø) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 9Ø).

POUR JOUER :
On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par Electron dès que l'on aura frappé RETURN.
Pour recommencer, il suffira de frapper la touche RETURN et ESCAPE pour finir.

EXTENSIONS POSSIBLES :
La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en ajoutant la couleur. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel Electron deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel Electron vous demande de trouver le mot qui correspond à un certain dessin, etc.

```
   10 REM VISIONS
   20 CLS
   30 INPUT"QUEL DESSIN : "R$
   35  COLOUR 129
   40 FOR I=1 TO LEN(R$)
   50   N=9*ASC(MID$(R$,I,1))-340
   60   J=1
   70   S=N/2-INT(N/2)
   80   N=INT(N/2)
   90   IF S<>0 THEN PRINTTAB(J+15,I+5);
" "
  100   J=J+1
  110   IF J<10 THEN 70
  120   NEXT I
  130 COLOUR 128:PRINT:PRINT
  140 INPUT"TAPER 'RETURN' "R$
  150 GOTO 20
>VDU12,3
```

421

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : moyenne.

LE JEU :
Chacun des joueurs devra, à son tour, lancer 3 dés, et peut ensuite choisir, sur les deux coups suivants, s'il désire relancer ou non 1, 2 ou 3 dés. Il marque alors un score correspondant au nombre indiqué par les dés, suivant les règles choisies, et le joueur suivant procède de la même manière.

LE PROGRAMME :
Le nombre de joueurs est tout d'abord entré en ligne 4Ø.
La ligne 13Ø affiche le numéro de joueur et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 4ØØØ, utilisé 3 fois dans la boucle J (2ØØ-25Ø), réalise le lancer de dés.
Les retirages sont effectués par les lignes 1ØØØ, 2ØØØ et 3ØØØ.
La variable R$ gère les 2ᵉ et 3ᵉ tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 14Ø-15Ø.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 26Ø et 28Ø.

POUR JOUER :
On lance l'exécution du programme par RUN ce qui conduit à l'affichage NOMBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape RETURN. L'écran s'efface et affiche JOUEUR Nº 1. ESSAI A, PARTIE P, ainsi que le tirage réalisé, et les noms des dés (A, B, C).
Si l'on ne désire pas refaire de tirage, il suffit de frapper RETURN. La main passe alors au joueur suivant.
Pour effectuer un autre tirage, il suffit de taper le nom du ou des dés à retirer, et de faire RETURN : par exemple, pour retirer A et C on tapera AC suivi de RETURN.
Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :
La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

```
  10 REM 421
  20 CLS
  40 INPUT"NOMBRE DE JOUEURS : "N
  50 P=1
  60 I=1
  70 K=0
  80 CLS
  90 R$="ABC"
 100 GOTO200
 110 K=K+1
 120 PRINTTAB(17,1):PRINT"-  4 2 1  -"
 130 PRINTTAB(0,4):PRINT"JOUEUR N.";I;"
 -  ESSAI N.";K;"  -  PARTIE N.";P
 140 PRINTTAB(7,7);A:PRINTTAB(17,7);B:P
RINTTAB(27,7);C
 150  PRINTTAB(7,9);"A":PRINTTAB(17,9);
"B":PRINTTAB(27,9);"C"
 160 R$="":PRINTTAB(0,14);"
     "
 170 INPUT TAB(0,14);"---    ABC : "R$
 180 IF R$="" THEN 270
 190 R$=R$+"DDD"
 200 FOR J=1 TO 3
 210   GOSUB4000
 220   X$=MID$(R$,J,1)
 230   X=ASC(X$)-64
 240   ON X GOSUB 1000,2000,3000,4000
 250   NEXT J
 260 IF K<3 THEN 110
 270 I=I+1
 280 IF I<=N THEN 70
 290 P=P+1
 300 GOTO 60
1000  A=D
1010 RETURN
2000  B=D
2010 RETURN
3000  C=D
3010 RETURN
4000  D=RND(6)
4010 RETURN
>VDU12,3
```

L'électricien fou

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Un immeuble comportant 1Ø fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer l'un des interrupteurs, en indiquant son numéro (de Ø à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 1Ø fenêtres, avec le moins possible d'essais.

LE PROGRAMME :
Une table A, en ligne 6Ø, contiendra l'état (allumé ou éteint) des 1Ø fenêtres.
Les lignes 3Ø à 5Ø et la boucle I (11Ø-16Ø) dessinent l'immeuble dans son état initial.
La ligne 17Ø regarde si tout est éteint, auquel cas l'on ira en ligne 31Ø afficher le nombre d'essais.
La ligne 18Ø attend l'indication de l'interrupteur à manœuvrer.
Les lignes 19Ø à 29Ø définissent le résultat de la manœuvre.

POUR JOUER :
RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTERRUPTEUR NUMERO, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de RETURN.
Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).
On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

EXTENSIONS POSSIBLES :
On pourra envisager un graphisme très amélioré avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

```
  10 REM L'ELECTRICIEN FOU
  20 CLS
  30 PRINTTAB(1,2);" -    -    -    -    -
 -    -    -    -    -"
  40 PRINTTAB(1,4);" -    -    -    -    -
 -    -    -    -    -"
  50 PRINTTAB(1,3);"¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦"
  60 M=0:DIM A(10)
  70 FORI=1TO7
  90   NEXTI
 100 T=0
 110 FORI=1TO10
 120   T=T+A(I)
 130   PRINTTAB(4*I-2,3);" "
 140   IF A(I)=0 THENPRINTTAB(4*I-2,3);
"#"
 150   PRINTTAB(4*I-2,6);I-1
 160   NEXTI
 170  IF T=10 THEN 310
 180  INPUTTAB(0,14);"--  INTERRUPTEUR
N. : "N
 190 N=N+1
 200 M=M+1
 210 IF A(N)=1 THEN 290
 220 A(N)=1
 230 R=TAN(RND(1)+N/RND(1)-N)-SIN(RND(1
)/N)+336*SIN(8*N)
 240 N=INT(10*(R-INT(R)))
 250 IF N=0 THEN N=RND(10)
 260 IF A(N)=1 THEN290
 270 A(N)=1
 280 GOTO 100
 290 A(N)=0
 300 GOTO230
 310 PRINT:PRINT"TROUVE EN ";M;" ESSAIS
."
>
```

La course au 20

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (deux joueurs, ou plus).
Difficulté du jeu : moyenne.

LE JEU :
Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 2Ø étoiles vont apparaître en début de partie, et chaque joueur peut prendre autant d'étoiles qu'il le désire ; mais ceci dans un seul tas à la fois. Celui qui prend la dernière étoile est considéré comme perdant.

LE PROGRAMME :
La boucle I (3Ø-5Ø) dessine 4 rangées de 2Ø étoiles.
La variable F, dimensionnée en ligne 7Ø, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées (ligne 6Ø).
La ligne 12Ø va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.
La boucle J (16Ø-18Ø) suppriment les étoiles ; la ligne 22Ø teste la fin de partie.

POUR JOUER :
Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JOUEUR 1 et LIGNES, puis CASES. On répond en entrant un numéro de ligne (entre 1 et 4) suivi de RETURN, et le nombre d'étoiles à enlever suivi de RETURN. Les étoiles sont alors supprimées, et c'est au tour du deuxième joueur.
La partie se termine, lorsque la dernière étoile a été enlevée, par l'affichage du joueur gagnant.

EXTENSIONS POSSIBLES :
Outre une amélioration de la qualité du graphisme et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est, les règles de départ : on peut autoriser l'enlèvement d'un nombre impair d'étoiles seulement, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...

```
   10 REM LA COURSE AU 20
   20 CLS
   30 FOR I=1 TO 4
   40   PRINTTAB(0,2*I);I;"   **********
**********"
   50   NEXTI
   60 T=0
   70 DIM F(4)
   80 B=1
   90 PRINTTAB(0,12);"JOUEUR N.";B
  100 PRINT:INPUT"LIGNE .... : "L
  110 PRINT:INPUT"CASES .... : "C
  120 IF C+F(L)>20 THEN 90
  130 F(L) = F(L)+C
  140 T=T+C
  160 FOR J=F(L)-C TO F(L)
  170   PRINTTAB(24-J,2*L);" "
  180   NEXTJ
  200 B=B+1
  210 IF B=3 THEN B=1
  220 IF T<80 THEN 90
  230 PRINT:PRINT"---  LE JOUEUR ";B;" A
 GAGNE ..."
>VDU12,3
```

Musique

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu éducatif.
Difficulté du jeu : moyenne.

LE JEU :
Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE, selon sa place sur la portée. Il faudra répondre en indiquant sur quelle ligne la note sera affichée.
La note est alors dessinée sur la portée, suivie d'un message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. Une deuxième boucle I (8Ø-12Ø) affiche les numéros des lignes.
La ligne 13Ø définit le nom des notes, et la ligne 14Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 15Ø et il est affiché par la ligne 16Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 19Ø.
La ligne 2ØØ s'occupe du décodage de la réponse.
La ligne 21Ø affiche la note.
Les lignes 24Ø et 25Ø regardent si l'on a gagné ou perdu et affichent le message correspondant.
La ligne 26Ø permet de faire un nouvel essai.

POUR JOUER :
On lance le programme par RUN.
On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question NUMERO DE LA LIGNE.
On répond en donnant un numéro de ligne (de 1 à E) suivi de RETURN.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On doit taper la touche RETURN pour continuer, et s'arrêter en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```
   10 REM MUSIQUE
   40 CLS
   50 FOR I=1TO5
   60   PRINTTAB(4,2*I+4);"------------
---------------------"
   70   NEXTI
   80 FORI=1TO14
   90   PRINTTAB(0,I+2);
  100   IF I<=9 THEN PRINT;I
  110   IF I>9 THEN PRINTCHR$(I+55)
  120   NEXTI
  130 N$="SI LA SOLFA MI RE DO "
  140 X=RND(14)
  150 T=(X-1) MOD 7
  160 PRINT TAB(0,18);"NOTE ";MID$(N$,3*
T+1,3);
  170 IF X<=7 THEN PRINT" AIGUE."
  180 IF X>7 THEN PRINT"GRAVE."
  190 PRINT:INPUT "NUMERO DE LA LIGNE :
";L$
  200 L= -(ASC(L$)-48)*(L$<":")-(ASC(L$)
-55)*(L$>="A")
  210 PRINT TAB(14,X);"!":PRINT TAB(14,X
+1);"!":PRINT TAB(13,X+2);"O"
  230 PRINT TAB(25,20)
  240 IF L=X THEN PRINT"===  BRAVO ..."
  250 IF L<>X THEN PRINT"===  PERDU ..."
  260 PRINT:INPUT"TAPER 'RETURN' POUR CO
NTINUER "R$
  270 GOTO40
>VDU12,3
```

Poker

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre Electron.
Difficulté du jeu : facile.

LE JEU :
C'est une simulation d'une partie de poker traditionnelle, effectuée avec un jeu de 32 cartes, dans laquelle le joueur aura pour partenaire Electron. Le joueur recevra 5 cartes, repérées par des numéros de 1 à 5, et pourra décider de changer certaines cartes en annonçant celles qu'il écarte. Sa nouvelle donne sera alors affichée, et c'est ensuite au tour de l'Electron de jouer.

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent les cartes. Une variable N est dimensionnée à 16, en ligne 4Ø, pour recevoir 1Ø cartes joueur et 5 cartes Electron.
Le tirage sera effectué par une boucle I (8Ø-14Ø) et le contrôle par une boucle J (11Ø-13Ø).
L'affichage des cartes est réalisé par le sous-programme 5ØØ-61Ø.
Une autre boucle I (19Ø-21Ø) réalise le deuxième tirage du joueur, la ligne 24Ø permettant le deuxième affichage, et la boucle I (5ØØ-57Ø) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :
L'ordre RUN lance l'exécution du programme, et les 5 premières cartes du joueur sont alors affichées.
A la question CARTES REFUSEES, le joueur peut répondre Ø suivi de RETURN s'il veut arrêter la partie. Pour jouer, il répondra simplement RETURN s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de RETURN.
Par exemple, s'il tape 235 et RETURN, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui de l'Electron, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire RETURN pour démarrer une autre partie.

EXTENSIONS POSSIBLES :
Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu de l'Electron, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```
   10 REM POKER
   40 DIMN(16)
   50 V$="7 8 9 10VADAROAS"
   60 C$="TRCACOPI"
   70 CLS
   80 FORI=1TO16
   90   N(I)=RND(32)-1
  100   IF I=1 THEN 140
  110   FOR J=1TOI-1
  120     IF N(I)=N(J)THEN J=I:NEXTJ:GOT
O90
  130   NEXTJ
  140 NEXTI
  150 K=0:P=0
  160 PRINT TAB(0,12);"CARTES REFUSEES :
 "
  170 GOSUB 500
  180 IF R$="0" THEN END
  190 FOR I=1 TO LEN(R$)
  200   N(VAL(MID$(R$,I,1)))=N(I+5)
  210   NEXTI
  220 CLS
  230 P=1
  240 GOSUB500
  250 K=K+10
  500 FORI=1TO5
  510   J=2*((N(I+K) MOD 4) +1)-1
  520   F=2*((N(I+K) MOD 8) +1)-1
  530   R$=MID$(V$,F,2)
  540   PRINT TAB(7*I-4,K);I
  550   PRINT TAB(7*I-5,K+2);R$;":";MID$
(C$,J,2)
  560   PRINT TAB(7*I-5,K+3);"-----"
  570   NEXTI
  580 IF P=1 THEN P=0:RETURN
  590 IF K=10 THEN PRINT:INPUT"TAPER 'RE
TURN' POUR CONTINUER : ";R$:GOTO70
  600 INPUT TAB(18,12);R$
  610 RETURN
>VDU12,3
```

Le jeu de saute-mouton

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

LE JEU :
Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons noirs et de 3 moutons gris, représentés par des symboles graphiques, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles pourront intervenir, selon votre propre estimation de l'intérêt de telle ou telle : soit l'on contraint les moutons à se placer sur la première case disponible, en sautant éventuellement par-dessus les moutons adverses, soit on leur laisse choisir leur case d'arrivée, ce qui peut rendre les parties interminables, à vous de choisir.

LE PROGRAMME :
La boucle I (5Ø-8Ø) initialise une table A (I), qui contient les positions des moutons. La boucle I (1ØØ-15Ø) dessine le terrain de jeu et les moutons.
Les lignes 17Ø et 18Ø s'occupent du texte et des entrées.
La ligne 19Ø contrôle la validité du coup. Le changement de joueur est réalisé en ligne 22Ø.

POUR JOUER :
On commence l'exécution du programme par RUN. On obtient alors un affichage JOUEUR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN.
On répond en indiquant le numéro du pion à déplacer suivi d'une virgule, puis le numéro de la case d'arrivée, suivi de RETURN.
Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.

EXTENSIONS POSSIBLES :
On commencera par rajouter un contrôle du joueur gagnant.
On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée.
Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

```
   10  REM LE JEU DE SAUTE-MOUTON
   20 CLS
   30 DIMA(8)
   40 A(1)=0:A(8)=0
   50 FORI=1TO3
   60   A(I+1)=1
   70   A(I+4)=-1
   80   NEXTI
   90 H=1
  100 FORI=1TO8
  110   PRINT TAB(4*I,8);I
  120   PRINT TAB(4*I,6);"  "
  130   IF A(I)=1 THENPRINT TAB(4*I,6);"
>"
  140   IF A(I)=-1 THENPRINT TAB(4*I,6);
"<"
  150   NEXTI
  160 J=1-(H=-1)
  170  PRINT TAB(0,13);"JOUEUR N. ";J
  180 PRINT:INPUT"DEBUT , FIN ...... : "
D,F
  190 IF A(D)<>H OR A(F)<>0 THEN 170
  200 A(D)=0
  210 A(F)=H
  220 H=-H
  230 GOTO100
>VDU12,3
```

Lettres

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). Electron choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :
La boucle I (9Ø-12Ø) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 6Ø. Pour éviter d'avoir trop de consonnes, on utilise en ligne 11Ø une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 13Ø-18Ø demande à chacun des joueurs la longueur de son mot. Les lignes 19Ø et 2ØØ comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 21Ø.

POUR JOUER :
On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.
Pour continuer la partie, il suffira d'appuyer sur la touche RETURN.
On devra faire ESCAPE pour finir.

EXTENSIONS POSSIBLES :
On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.
On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre Electron, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard ?

```
   10 REM LETTRES
   40 S=0:J=0
   50 CLS
   60 L=RND(6)+6
   70 A$="AEIOUEA"
   80 PRINT TAB(0,6);"LETTRES :"
   90 FOR I=1 TO L
  100   PRINT TAB(8+2*I,6);CHR$(RND(26)+
64)
  110   IF RND(1)<.4 THEN PRINT TAB(8+2*
I,6);MID$(A$,RND(7),1)
  120   NEXTI
  130 I=1
  140 PRINT TAB(0,10);"JOUEUR N. ";I;:INP
UT" -- LONGUEUR .... : "L
  150 PRINT TAB(25,10);"    "
  160 IF I=1 THEN P=L
  170 I=I+1
  180 IF I<3 THEN 140
  190 S=S+(P AND P>L)
  200 J=J+(L AND L>P)
  210 PRINT TAB(0,13);"SCORES : JOUEUR 1
 : ";S;" -- JOUEUR 2 : ";J
  220 PRINT:INPUT"TAPER 'RETURN' POUR CO
NTINUER : "R$
  230 GOTO 50
>VDU12,3
```

Espion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : facile.

LE JEU :
Le programme peut servir simultanément de codeur et de décodeur de message secret, suivant une méthode très facile à mettre en œuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien veiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME :
Le noyau initial (qui sera un entier quelconque) est entré en ligne 3Ø ; le message est entré à la ligne 5Ø.
La ligne 6Ø permet de choisir entre les options de codage et de décodage.
La boucle I (8Ø-15Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 1ØØ appelle le générateur aléatoire.
La ligne 11Ø réalise la transposition des lettres.
Les lignes 12Ø et 13Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 16Ø.
Les lignes 1ØØØ et 1Ø3Ø contiennent le générateur.

POUR JOUER :
Lancer le programme par RUN.
Entrer ensuite un entier N qui servira de noyau pour le générateur (il faudra le retenir pour le décodage).
Puis taper le message, suivi de RETURN.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper RETURN.
Le message codé ou décodé, selon le cas, sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra étendre le codage à d'autres signes que les lettres de l'alphabet, utiliser éventuellement plusieurs générateurs, introduire dans le message un codage du ou des nombres ayant servi de générateurs, etc.

```
   10 REM ESPION
   20 CLS
   30 INPUT"TAPEZ VOTRE CODE : "N
   40 A=1/N
   50 PRINT:INPUT"VOTRE MESSAGE : "A$
   60 PRINT:INPUT"VOULEZ-VOUS (C)ODER OU
 (D)ECODER ? : "R$
   70 B$=""
   80 FOR I=1 TO LEN(A$)
   90   T=ASC(MID$(A$,I,1))
  100   GOSUB1000
  110   T=T-A*((R$="C")-(R$="D"))
  120   IF T>90 THEN T=T-26
  130   IF T<65 THEN T=T+26
  140   B$=B$+CHR$(T)
  150   NEXT
  160 PRINT:PRINT"RESULTAT : ";B$:END
  999 END
 1000 A=(A+3.14159)^2
 1010 A=A-INT(A)
 1020 A=INT(26*A+1)
 1030 RETURN
>VDU12,3
```

Accrochez les wagons

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Il faut essayer d'accrocher, le plus rapidement possible, la locomotive à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).
On tapera, par exemple : BBGDHHH.

LE PROGRAMME :
Les lignes 4Ø et 5Ø dessinent la locomotive, la ligne 6Ø le wagon, et les roues sont en ligne 7Ø.
La position initiale de la locomotive est définie par les lignes 9Ø et 1ØØ, celle du wagon par les lignes 11Ø et 12Ø.
La ligne 13Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 15Ø et 16Ø réalisent l'affichage du wagon et les lignes 17Ø, 18Ø et 19Ø celui de la locomotive.
Le test de fin est en ligne 2ØØ.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 21Ø, et sont décodés dans une boucle I (24Ø-27Ø) pour être exécutés en ligne 28Ø.

POUR JOUER :
RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivie de RETURN.
La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.
On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :
Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.
On pourra aussi prévoir plusieurs wagons et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

```
  10 REM ACCROCHEZ LES WAGONS
  40 D$="  *"
  50 A$="}=!"
  60 B$="!_!"
  70 C$="O O"
  80 N=0
  90 LY=RND(33)-1
 100 LX=RND(20)
 110 WY=RND(33)-1
 120 WX=RND(20)
 130 IF ABS(LX-WX)<4 THEN 90
 140 CLS
 150 PRINT TAB(WY,WX);B$
 160 PRINT TAB(WY,WX+1);C$
 170 PRINT TAB(LY,LX-1);D$
 180 PRINT TAB(LY,LX);A$
 190 PRINT TAB(LY,LX+1);C$
 200 IF LX=WX AND WY=LY-3 THEN 300
 210 INPUT TAB(0,22);"CHEMIN A SUIVRE (
DGGHBB...) : "R$
 220 X=0:Y=0
 230 N=N+1
 240 FOR I=1 TO LEN(R$)
 250   X=X-(MID$(R$,I,1)="B" AND LX<22)
+(MID$(R$,I,1)="H" AND LX >0)
 260   Y=Y-(MID$(R$,I,1)="D" AND LY<37)
+(MID$(R$,I,1)="G" AND LY >0)
 270   NEXT I
 280 LX=LX+X:LY=LY+Y
 290 GOTO 140
 300 PRINT TAB(0,22);"=== BRAVO. TROUVE
 EN ";N;" COUPS..."
>VDU12,3
```

Tirage d'un dé truqué

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.

LE PROGRAMME :
Les probabilités désirées pour chaque tirage sont introduites par une boucle I (7Ø-14Ø)avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 16Ø.
La boucle J (17Ø-25Ø) comptabilise les tirages.
La boucle I (19Ø-21Ø) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 22Ø, et affichés grâce aux lignes 23Ø, 24Ø et 25Ø.

POUR JOUER :
Commencer l'exécution en frappant RUN, puis à la question PROBABILITE DU DE 1, introduire un nombre compris entre Ø et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de RETURN. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.
Les résultats sont affichés sous les numéros des tirages : ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de Ø.3, a été réalisé 32 fois.

EXTENSIONS POSSIBLES :
On peut envisager d'améliorer la représentation des résultats en utilisant par exemple un histogramme, ou toute autre méthode graphiquement plus parlante.

```
   10 REM TIRAGE D'UN DE TRUQUE
   40 DIM E(6)
   50 CLS
   60 P=0
   70 FOR I=1 TO 6
   80   PRINT TAB(0,3);STRING$(40," ")
   90   PRINT TAB(0,3);"PROBABILITE DU D
E ";I;:INPUT" : "E(I)
  100   P=P+E(I)
  110   IF E(I)<0 OR E(I)>1 OR P>1 THEN
300
  120   PRINT TAB(6*I-6,6);"DE ";I
  130   PRINT TAB(6*I-5,8);E(I)
  140   NEXTI
  150 IF P<.9 THEN 50
  160 INPUT TAB(0,11);"NOMBRE DE TIRAGES
 .... : "P
  170 FOR J=1 TO P
  180   N=RND(6)
  190   FOR I=1 TO 6
  200     IF N=I AND RND(1)>E(I)-INT(E(I
)) THEN I=6:NEXTI:GOTO 180
  210   NEXT I
  220 E(N)=E(N)+1
  230 PRINT TAB(6*N-4,16);N
  240 PRINT TAB(6*N-5,18);E(N)
  250 PRINT TAB(25,11);P-J;"    "
  260 NEXT J
  270 END
  300 PRINT TAB(0,4);"=== IMPOSSIBLE : 0
<=P<=1"
  310 P=P-E(I)
  320 FOR J=1 TO 1700:NEXT J
  330 PRINT TAB(0,4);STRING$(40," ")
  340 GOTO 90
>VDU12,3
```

Tour de cartes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.

LE JEU :
C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par Electron qui tient le rôle du prestidigitateur ; 21 cartes, prises au hasard dans un paquet de 52, sont affichées, et le joueur devra retenir mentalement l'une des cartes présentées. Puis, il devra indiquer à l'Electron dans quelle colonne (1re, 2e ou 3e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4e affichage, Electron est en mesure de deviner la carte choisie.

LE PROGRAMME :
La boucle I (6Ø-12Ø) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (8Ø-1ØØ) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 14Ø contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (15Ø-24Ø) réalise l'affichage du jeu.
La ligne 16Ø décode la couleur de la carte.
La ligne 17Ø choisit la position d'affichage.
La ligne 18Ø décode la hauteur.
L'affichage proprement dit est effectué de la ligne 19Ø à la ligne 23Ø.
La ligne 26Ø permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (3ØØ-32Ø) réalise le brouillage, et les boucles emboîtées J (35Ø-39Ø) et I (36Ø-38Ø) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

POUR JOUER :
On lance le programme par RUN.
On obtient un premier affichage de 21 cartes réparties sur 3 colonnes.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de RETURN.
On obtient un 2e affichage et l'on fait de même ; puis un 3e affichage et l'on fait de même.
Au 4e affichage, la carte choisie est automatiquement affichée.
On sortira du programme en tapant ESCAPE à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :
On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur. On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
   10 REM TOUR DE CARTES
   40 CLS
   50 A$=""
   60 FOR I=1 TO 21
   70    B$=CHR$(RND(52)+37)
   80    FOR J=1 TO LEN(A$)
   90       IF B$=MID$(A$,J,1) THEN J=LEN(
A$):NEXTJ:GOTO70
  100    NEXTJ
  110 A$=A$+B$
  120 NEXTI
  130 H=3:P=1
  140 B$="TRCACOPIVADAROAS"
  150 FOR I=1 TO LEN(A$)*P-11*(P=0)
  160    S=INT((ASC(MID$(A$,I,1))-38)/13)
  170    J=7*INT((I-1)/7)
  180    T=ASC(MID$(A$,I,1))-37-13*S
  190    IF P=0 THEN PRINT TAB(0,0);"VOTR
E CARTE EST :                 "
  200    PRINT TAB(J*P+INT(J/2)-16*(P=0),
(I-J)*P);
  210    IF T<=9 THENPRINTSTR$(T+1);
  220    IF T>9 THEN PRINTMID$(B$,ABS(2*T
-11),2);
  230    PRINT TAB((J+3)*P+1+INT(J/2)-21*
(P=0),(I-J)*P);MID$(B$,2*S+1,2)
  240    NEXTI
  250 IF P=0 THEN END
  260 INPUT TAB(0,16);"QUELLE COLONNE :
"N
  270 CLS
  280 H=H-1
  290 R$="20120":B$=""
  300 FOR J=0 TO 2
  310    B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,
1))+1,7)
  320    NEXTJ
  330 IF H=0 THEN A$=B$:P=0:GOTO 140
  340 A$=""
  350 FOR J=0 TO 2
  360    FOR I=1 TO 7
  370       A$=A$+MID$(B$,3*I-2+J,1)
  380       NEXT I
  390    NEXT J
  400 GOTO 140
>VDU12,3
```

Le jeu de Marienbad

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : logique (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines, à savoir le jeu de NIM.
En effet, il est possible, en début de partie, de choisir le nombre d'allumettes que l'on veut mettre dans chaque tas.
Chaque joueur joue à son tour, et celui qui retire la dernière allumette a perdu.

LE PROGRAMME :
La ligne 60 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (70-100) permet d'entrer ce nombre.
La boucle I (130-160) imprime les tas.
La ligne 170 assure l'affichage du joueur.
La ligne 190, ainsi que la ligne 200, contrôlent si le coup est autorisé.
La ligne 270 contrôle la fin de la partie.

POUR JOUER :
On lance le programme avec RUN, puis on entre le nombre de tas (entre 1 et 7), suivi de RETURN.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de RETURN). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message DANS QUEL TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de RETURN. On obtient ensuite la question QUANTITE A ENLEVER, et l'on répond en indiquant le nombre d'allumettes que l'on désire enlever du tas concerné.
Puis ce sera au tour du deuxième joueur, et l'on jouera alternativement jusqu'à ce qu'il n'y ait plus d'allumettes disponibles.
En fin de partie, on obtiendra l'affichage du numéro du joueur gagnant.

EXTENSIONS POSSIBLES :
On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant à Electron de servir de partenaire, etc.

```
   10 REMLE JEU DE MARIENBAD
   20 CLS
   30 INPUT TAB(0,1);"NOMBRE DE TAS ?
(=7 ) : " N
   40 IF N)=8 OR N(1 THEN 30
   50 S=0
   60 DIM A(N)
   70 FOR I=1 TO N
   80   PRINT TAB(0,I*3+2);"TAS N.";I;:I
NPUT "  -  NOMBRE D'ALLUMETTES : " A(I)
   90   S=S+A(I)
  100   NEXT I
  110 H=1
  120 CLS
  130 FOR I=1 TO N
  140   PRINT TAB(5*I+2,4);A(I)
  150   PRINT TAB(0,6);"TAS :":PRINT TAB
(5*I+2,6);I
  160   NEXT I
  170 PRINT TAB(0,10);"JOUEUR N.";H
  180 INPUT TAB(0,12);"DANS QUEL TAS : "
 T
  190 IF T)N OR T(1 THEN 120
  200 IF A(T)=0 THEN 120
  210 INPUT TAB(0,14);"QUANTITE A ENLEVE
R : " O
  220 IF O)A(T) OR O(1 THEN 200
  230 A(T)=A(T)-O
  240 S=S-O
  250 H=H+1
  260 IF H=3 THEN H=1
  270 IF S=0 THEN PRINT TAB(0,19);CHR$(7
);"LE JOUEUR N.";H;" A GAGNE ...":END
  280 GOTO120
)VDU12,3
```

L'indésirable

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard et réflexion (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes le propriétaire d'un immeuble de 16 pièces vides, et vous n'arrivez pas à mettre la main sur votre locataire indésirable. Vous visitez les pièces au hasard, mais, si vous visitez la pièce voisine, votre locataire, prévenu par le bruit, va déménager. Il peut même se réinstaller dans une pièce que vous avez déjà visitée, et marquée comme vide. Il vous faudra arriver à le déloger en moins de 9 coups pour gagner.

LE PROGRAMME :
Une table B est dimensionnée à 16, en ligne 6Ø, pour recevoir les numéros des cases déjà visitées.
La boucle I (8Ø-11Ø) dessine le terrain de jeu.
La position initiale du locataire est tirée au hasard en ligne 12Ø.
Le numéro de la case à visiter est donné en ligne 13Ø.
Les lignes 14Ø à 22Ø contrôlent les diverses réponses possibles.
La boucle I (18Ø-2ØØ) dessine les cases visitées.
La ligne 23Ø dessine le locataire, effacé en ligne 26Ø.
Les lignes 27Ø et 28Ø assurent son déplacement.

POUR JOUER :
On lance l'exécution par RUN.
A la question CASE VISITEE, on répond en tapant une lettre, de A à P. Si on a trouvé le locataire, celui-ci est affiché, et la partie est terminée ; sinon, deux cas peuvent se produire : soit l'on est suffisamment loin, et le locataire ne bouge pas, soit l'on est à une case, il apparaît un court instant, et il se recache, aléatoirement, dans une case voisine. Les cases déjà visitées se colorent en rouge, et vous ne pouvez pas y revenir. Si, au bout de 9 coups, vous n'avez pas trouvé, la position du locataire s'affiche, et vous avez perdu.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, jouer sur plus de 16 cases, utiliser des caractères graphiques pour représenter le locataire et le propriétaire, utiliser un algorithme plus subtil pour le déplacement du locataire, lui interdire éventuellement de revenir dans des cases déjà visitées, etc.

```
   10 REM L'INDESIRABLE
   20 MODE 1
   40 A$="#"
   50 B$=A$+A$+A$
   60 DIM B(16)
   70 CLS
   80 FOR I=1 TO 16
   90    COLOUR 1:PRINT TAB(4+4*((I-1) MO
D 4),2+4*INT((I-1)/4));B$
  100    COLOUR 1:PRINT TAB(4+4*((I-1) MO
D 4),3+4*INT((I-1)/4));A$;:COLOUR 3:PRIN
T CHR$(64+I);:COLOUR 1:PRINTA$
  110    NEXT I
  120 X=RND(16):S=1:K=0
  130 COLOUR 3:INPUT TAB(0,20);"CASE VIS
ITEE : " C$:C=ASC(C$)-64
  140 IF B(C)=1 THENPRINT TAB(0,22);"===
  DEJA VISITEE ...":GOTO130
  150 IF C=X THEN PRINT TAB(0,22);CHR$(7
);"VOUS L'AVEZ EU EN ";S;" COUPS. BRAVO
...":K=1:GOTO230
  160 IF X-5=C OR X-4=C OR X-3=C OR X-1=
C OR X+1=C OR X+3=C OR X+4=C OR X+5=C TH
EN 230
  170 PRINT TAB(0,22);"... IL N'EST PAS
LA !"
  180 S=S+1:B(C)=1:FOR I=1 TO 16
  190    COLOUR 1+B(I):PRINT TAB(4+4*((I-
1) MOD 4),2+4*INT((I-1)/4));B$
  200    NEXT I
  210 IF S=9 THEN PRINT TAB(0,22);"VOUS
AVEZ PERDU. IL ETAIT EN ";CHR$(X+64):K=1
:GOTO230
  220   PRINT TAB(0,22);STRING$(40," "):G
OTO130
  230   COLOUR 2:PRINT TAB(5+4*((X-1) MOD
 4),1+4*INT((X-1)/4));"%"
  240 FOR I=1 TO 300:NEXT I
  250   IF K=1 THEN PRINTCHR$(7);:COLOUR
3:END
  260 COLOUR 3:PRINT TAB(5+4*((X-1) MOD
4),1+4*INT((X-1)/4));" "
  270 Z=RND(10):IF ABS(Z)=2 THEN 270
  280 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GO
TO 270
  290 GOTO180
>VDU12,3
```

Compose

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : composition musicale.
Difficulté du jeu : aucune, a priori.

LE JEU :
Il s'agit, en fait, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur nom. Une nouvelle série de notes sera ensuite affichée, et ainsi de suite.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. La variable N$, en ligne 8Ø, contient les noms des notes.
La deuxième boucle I (9Ø-17Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 1ØØ ; son nom est calculé en ligne 11Ø et la note est dessinée par la ligne 12Ø.
La ligne 18Ø permet de recommencer en tapant RETURN.

POUR JOUER :
Lancer l'exécution du programme en tapant RUN.
La portée sera alors rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur la touche RETURN provoquera l'affichage d'une nouvelle série de notes, et une pression sur les touches ESCAPE provoque la fin de la partie.

EXTENSIONS POSSIBLES :
On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme, faire jouer la mélodie affichée.

```
   10 REM COMPOSE
   40 CLS
   50 FOR I=1 TO 5
   60    COLOUR 2:PRINT TAB(0,3+2*I);STR
ING$(40,"-")
   70   NEXTI
   80 N$="DO SI LA SOLFA MI RE "
   90 FOR I=1 TO 13
  100   X=RND(14)
  110   D$=MID$(N$,3*(X MOD 7)+1,3)
  120   COLOUR 1:PRINT TAB(3*I-2,X-1);"!
":PRINT TAB(3*I-2,X);"!":PRINT TAB(3*I-3
,X+1);"O"
  130   COLOUR 3
  140   PRINT TAB(3*I-3,18);D$
  170   NEXTI
  180 INPUT TAB(0,21);"TAPER 'RETURN' PO
UR CONTINUER : "R$
  190 GOTO 40
>VDU12,3
```

Jour de la semaine

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : utilitaire.
Difficulté du jeu : facile.

LE JEU :
Le programme vous permet de retrouver, sans aucun calcul, le jour de la semaine correspondant à une date donnée.
La date sera entrée sous forme de trois nombres, par exemple 11, 3, 1956 pour le 11 mars 1956, et le résultat sera affiché en clair.

LE PROGRAMME :
Les lignes 3Ø à 7Ø contrôlent les entrées. La ligne 4Ø s'occupe de la fin de partie. La ligne 9Ø permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 1ØØ à 11Ø permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 13Ø et 14Ø.
Les lignes 15Ø et 16Ø permettront l'affichage en clair du jour dans la réponse. La ligne 17Ø imprime les résultats.
Le sous-programme 2ØØ-22Ø intervient pour tenir compte des mois de janvier et février.

POUR JOUER :
Lancer le programme par RUN, puis entrer le jour suivi de RETURN, le mois (entre 1 et 12) suivi de RETURN, et l'année (sous la forme 1845, par exemple) suivie de RETURN.
Le résultat est très rapidement affiché.
On tape RETURN et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année.
Pour terminer, il faut répondre Ø à la question JOUR.

EXTENSIONS POSSIBLES :
On pourra entrer les mois en clair et faire afficher le nom du jour en entier.
On peut envisager de faire afficher les dates et fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).
On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

```
   10 REM JOUR DE LA SEMAINE
   20 CLS
   30 INPUT TAB(0,1);"JOUR ........ : "
J
   40 IF J=0 THEN END
   50 INPUT TAB(0,3);"MOIS ........ : "
M
   60 INPUT TAB(0,5);"ANNEE ....... : "
A
   70 B=A
   80 S=INT(A/100)-INT(A/400)
   90 IF A<=1582 THEN S=2
  100 M$="JANVFEVRMARSAVRIMAI JUINJUILAO
UTSEPTOCTONOVEDECE"
  110 M$=MID$(M$,4*M-3,4)
  120 IF M<3 THEN GOSUB 200
  130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
  140 S=S-7*INT(S/7)
  150 J$="DIMANCHELUNDI   MARDI   MERCRE
DIJEUDI   VENDREDISAMEDI  "
  160 J$=MID$(J$,8*S+1,8)
  170 PRINT TAB(0,15);"LE ";J;" ";M$;" "
;B;" ETAIT UN ";J$
  180 INPUT TAB(0,19);"TAPER 'RETURN' PO
UR CONTINUER : " R$
  190 GOTO 20
  200 M=M+12
  210 A=A-1
  220 RETURN
>VDU12,3
```

Le jeu de la vie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu solitaire (2 versions).
Difficulté du jeu : facile.

LE JEU :
La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.
Une certaine population est définie au début du jeu soit par Electron (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :
Deux tables (B et C) sont utilisées en ligne 5Ø, pour enregistrer l'état de la population. Les boucles F (23Ø-31Ø) et G (24Ø-3ØØ) sont utilisées en début de jeu pour définir la population initiale (lignes 25Ø et 26Ø). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 8Ø-21Ø, 9Ø-2ØØ, 11Ø-16Ø, 12Ø-15Ø. La ligne 13Ø réalise la gestion du nouvel état suivant les règles.
L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (23Ø-31Ø) et G (24Ø-3ØØ).
La ligne 32Ø permet de recommencer.

POUR JOUER :
Dans la version automatique, il suffit de lancer le programme par RUN, d'indiquer la dimension du tableau et de regarder. Pour arrêter, il faudra taper ESCAPE. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant Ø si l'on ne veut rien mettre sur la ligne et la colonne indiquée, et 1 si l'on désire y mettre une cellule. Puis l'évolution se déroulera ensuite de façon automatique.

EXTENSIONS POSSIBLES :
Il sera facile, si l'on dispose d'une mémoire plus importante, d'augmenter les dimensions des tables B et C, en redéfinissant la variable A. On pourra aussi introduire la couleur, améliorer la précision du dessin en remplaçant les PRINT par un affichage haute résolution.

```
   10 REM LE JEU DE LA VIE (AUTOMATIQUE)
   30 CLS
   40 INPUT TAB(0,1);"DIMENSION ...... :
  " A
   50 DIM B(A,A),C(A,A)
   60 I=0
   70 GOTO 220
   80 FOR F=1 TO A
   90   FOR G=1 TO A
  100     S=0
  110     FOR I=-1 TO 1
  120       FOR J=-1 TO 1
  130         IF F+J=0 OR G+I=0 OR F+J=A
+1 OR G+I=A+1 THEN 150
  140         S=S+B(F+J,G+I)
  150         NEXT J
  160       NEXT I
  170     S=2*S-B(F,G)
  180     C(F,G)=0
  190     IF S>=5 AND S<=7 THEN C(F,G)=1
  200     NEXT G
  210   NEXT F
  220 CLS
  230 FOR F=1 TO A
  240   FOR G=1 TO A
  250     IF I<>0 THEN 270
  260     IF RND(1)<.5 THEN C(F,G)=1
  270     PRINT TAB(2*G,2*F);:IF C(F,G)=
1 THEN PRINT"*" ELSE PRINT"."
  280     B(F,G)=C(F,G)
  290     NEXT G
  300   NEXT F
  310 GOTO 80
>VDU12,3
```

```
  10 REM LE JEU DE LA VIE (MANUEL)
  30 CLS
  40 INPUT TAB(0,1);"DIMENSION ...... :
 " A
  50 DIM B(A,A),C(A,A)
  60 I=0
  70 GOTO 220
  80 FOR F=1 TO A
  90   FOR G=1 TO A
 100     S=0
 110     FOR I=-1 TO 1
 120       FOR J=-1 TO 1
 130         IF F+J=0 OR G+I=0 OR F+J=A
+1 OR G+I=A+1 THEN 150
 140         S=S+B(F+J,G+I)
 150         NEXT J
 160       NEXT I
 170     S=2*S-B(F,G)
 180     C(F,G)=0
 190     IF S>=5 AND S<=7 THEN C(F,G)=1
 200     NEXT G
 210   NEXT F
 220 CLS
 230 FOR F=1 TO A
 240   FOR G=1 TO A
 250     IF I<>0 THEN 280
 260     PRINT TAB(0,25);"== LIGNE ";F;
"    == COLONNE ";G:PRINT TAB(26,27);"TA
PER 1 OU 0"
 270     INPUT TAB(2*G,2*F);""C(F,G)
 280     PRINT TAB(2*G,2*F);:IF C(F,G)=
1 THEN PRINT"*" ELSE PRINT"."
 290     B(F,G)=C(F,G)
 300     NEXT G
 310   NEXT F
 320 GOTO 80
>VDU12,3
```

Niveau 3

Nom du programme	*Thèmes et particularités*
Machinamot	Laissez créer des vocables nouveaux
Un dé	Tirage d'un dé, avec graphisme.
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
A travers	Un jeu de stratégie pour 2 joueurs.
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Horloge 1	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Tours de Hanoï	Un conseil : commencez petit ! ! !
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.

Manipulation des chaînes L'instruction DATA

TRAITEMENT DES CHAINES DE CARACTERES : MID$, LEFT$, RIGHT$

Nous avons eu l'occasion de rencontrer, dès le niveau II, la fonction essentielle utilisée dans le Basic Electron pour manipuler les chaînes : il s'agit de MID$ (A$, I J) qui permet d'extraire, dans la chaîne A$, une sous-chaîne de longueur J, commençant au Ième caractère.

Ainsi si,

A$ = "MONSIEUR"

on aura :

MID$ (A$, 2, 3) = "ONS"

Pour faciliter les manipulations, il est possible d'utiliser les deux fonctions LEFT$ et RIGHT$.

LEFT$ (A$, I) permet d'extraire les I premiers caractères à partir de la gauche du mot A$.

Ainsi, LEFT$ (A$, 2) = "MO", avec l'exemple précédent (on pourrait d'ailleurs utiliser MID$ (A$, 1, 2) pour obtenir le même résultat).

RIGHT$ (A$, J) permet d'extraire les J derniers caractères (on part de la droite du mot A$).

Ainsi RIGHT$ (A$, 3) = "EUR", toujours avec le même exemple.

Il est plus difficile de se passer de la fonction RIGHT$, mais il serait possible de faire MID$ (A$, LEN (A$) - 3, 3) pour obtenir le même résultat.

Le Basic de l'Electron ne permet pas le remplacement direct d'une sous-chaîne par une affectation du type MID$ (-.-) = " ". Il sera nécessaire, pour remplacer la Ième lettre de la chaîne A$, d'écrire :

A$ = LEFT$ (A$, I − 1) + "nouvelle lettre" + MID$ (A$, I + 1).

Des formules analogues pourront être utilisées pour remplacer dans A$ une sous-chaîne par une autre, pas nécessairement de même longueur, ce qui permet une grande souplesse.

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

Pour remplir facilement un tableau, numérique ou alphanumérique, il sera intéressant d'utiliser des lignes DATA, qui contiennent les données inscrites les unes à la suite des autres, séparées par une virgule.

Ainsi, on pourra écrire :

DATA LEON, JULES, 13, rue MARTIN...

Ces données peuvent être lues dans des boucles FOR... NEXT, au moyen de l'instruction READ. On peut recommencer la lecture au début des DATA, à l'aide de l'instruction RESTORE.

Machinamot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de hasard (sur des mots).
Difficulté du jeu : aucune.

LE JEU :
Ce très court programme permet à Electron de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :
Deux variables A$ et B$ sont dimensionnées à 15 en ligne 5Ø et 6Ø : l'une contiendra les préfixes, l'autre les suffixes (si l'on désire rajouter des possibilités, il suffira de modifier cette limitation à 15).
La boucle I (7Ø-9Ø) lit les lignes DATA (16Ø-18Ø) concernant les préfixes suivis des suffixes.
La ligne 1ØØ effectue le choix d'un préfixe au hasard.
La ligne 11Ø effectue le choix d'un suffixe au hasard.
Le mot obtenu en ligne 12Ø est affiché par la ligne 13Ø, puis on recommence (ligne 15Ø).

POUR JOUER :
Lancer l'exécution du programme par RUN. Un premier mot sera immédiatement affiché, puis un nouveau mot sera affiché à la suite du précédent.
Taper ESCAPE pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra très facilement augmenter la liste des mots disponibles.
On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris à Electron...

```
   10  REM MACHINAMOT
   40 CLS
   50  DIM A$(15)
   60 DIM B$(15)
   70 FOR I=1 TO 15
   80   READ A$(I),B$(I)
   90   NEXT I
  100 I=RND(15)
  110 J=RND(15)
  120 C$=A$(I)+B$(J)
  130 PRINTC$
  140 FOR S=1 TO 1200 : NEXT
  150 GOTO100
  160 DATAANDRO,CRATE,COSMO,PHILE,MICRO,
PJOBE,HELIO,TROPE,SEXO,GYNE,STEREO,MANE
  170 DATAPALEO,NAUTE,MACRO,PHAGE,HYDRO,
FUGE,STENO,PATHE,PATHO,CARDE,OLEO,DERME
  180 DATAMYTHO,SCOPE,HOMEO,VORE,OCTO,PH
ONE
>VDU12,3
```

Un dé

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : tirage d'un dé (avec graphisme).
Difficulté du jeu : aucune.

LE JEU :
Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.
Il pourra facilement être utilisé en sous-programme, dans tout type de jeux faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :
La ligne 5Ø utilise le générateur aléatoire de l'Electron pour tirer un nombre de 1 à 6.
La boucle J (7Ø-9Ø) dessine les bords du dé, et les lignes 6Ø et 1ØØ dessinent le haut et le bas.
Les lignes 12Ø à 17Ø sont utilisées pour le dessin des points.
Les lignes 18Ø et 19Ø permettront le cas échéant de refaire un tirage.

POUR JOUER :
Lancer le programme par RUN.
Un dé s'affiche, en couleurs, et en grand format.
A la question : AUTRE TIRAGE (O/N) on répond O, suivi de ENTREE, pour relancer le dé.

EXTENSIONS POSSIBLES :
On pourra introduire une petite musique agrémentant chaque lancer, faire défiler les points comme si le dé roulait sur une table, etc.

```
   10 REM UN DE
   40 CLS
   50 A=RND(6)
   60 COLOUR 129:PRINT TAB(10,5);STRING$
(13," ")
   70 FOR J=0 TO 10
   80   PRINT TAB(10,6+J);" ":PRINT TAB(
22,6+J);" "
   90   NEXTJ
  100 PRINT TAB(10,17);STRING$(13," ")
  120 IF A<>1 THENPRINT TAB(13,8);" "
  130 IF A>=4 THENPRINT TAB(19,8);" "
  140 IF A=6 THENPRINT TAB(13,11);" ":PR
INT TAB(19,11);" "
  150 IF (A MOD 2)=1 THEN PRINT TAB(16,1
1);" "
  160 IF A>=4 THEN PRINT TAB(13,14);" "
  170 IF A<>1 THEN PRINT TAB(19,14);" "
  180 COLOUR 128:INPUT TAB(0,22);"AUTRE
TIRAGE ? (O/N) : " R$
  190 IF LEFT$(R$,1)="O" THEN 40
>VDU12,3
```

Plus grand, plus petit

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 7 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3Ø54, et que l'on essaie 4251, la machine répondra 2 plus grands (ce sont le 4 et le 2), et 1 plus petit (c'est le 1).

LE PROGRAMME :
La ligne 6Ø interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé en ligne 8Ø, dans une variable alphanumérique X$.
Les comparaisons avec le nombre essayé sont faites à la ligne 13Ø (qui teste si l'on a trouvé juste), et dans la boucle I (15Ø-18Ø) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 19Ø.
Le compteur du nombre d'essais est en ligne 2ØØ.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage NOMBRE DE CHIFFRES.
On entre un nombre inférieur ou égal à 8 et on appuie sur la touche RETURN.
Le numéro de l'essai s'affiche, ainsi que la question NOMBRE ESSAYE ; on répond en entrant un nombre, suivi de RETURN.
Le résultat est alors affiché sous la forme > et <.
S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).
Si l'on a trouvé Electron affichera VOUS AVEZ TROUVE... EN... ESSAIS. Puis AUTRE PARTIE (O/N) ; on fera O suivi de RETURN pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :
On pourra, comme pour le Mastermind, imaginer un jeu analogue avec des lettres, des couleurs, ou même en utilisant d'autres symbolismes.
On peut aussi prévoir de laisser jouer Electron qui tiendra alors le rôle de décodeur.

```
   10 REM PLUS GRAND, PLUS PETIT
   40 CLS
   50 INPUT TAB(0,1);"NOMBRE DE CHIFFRES
 : " C
   60 IF C<1 OR C>8 THEN 40
   70 E=1:F=1
   80 X$=STR$(RND(10^C))
   90 X$=RIGHT$("00000000"+X$,C)
  100 PRINT TAB(0,3);"ESSAI N.   NOMBRE
ESSAYE   >    <"
  110 PRINT TAB(0,4);STRING$(40,"-")
  120 PRINT TAB(4,F+4);E:INPUT TAB(13,F+
4);"" P$
  130 IF P$=X$ THEN 220
  140 A=0:B=0
  150 FOR I=1 TO C
  160   IF MID$(P$,I,1)>MID$(X$,I,1) THE
N A=A+1
  170   IF MID$(P$,I,1)<MID$(X$,I,1) THE
N B=B+1
  180   NEXT I
  190 PRINT TAB(27,F+4);A:PRINT TAB(31,F
+4);B
  200 E=E+1:F=F+1:IF F=18 THEN F=1
  210 GOTO 120
  220 PRINT TAB(0,26);"VOUS AVEZ TROUVE
";X$;" EN ";E;" ESSAIS."
  230 INPUT TAB(0,28);"AUTRE PARTIE ? (O
/N) : " R$
  240 IF LEFT$(R$,1)="O" THEN 40
>VDU 12,3
```

Ordre

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Les 1Ø premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

```
B A D G H I J C F E
Ø 1 2 3 4 5 6 7 8 9   : permutation 83

B A H F D J I E G C
Ø 1 2 3 4 5 6 7 8 9   : résultat
```

LE PROGRAMME :
Les boucles emboîtées I (6Ø-12Ø) et J (8Ø-1ØØ) définissent une variable alphanumérique A$, contenant les 1Ø premières lettres de l'alphabet dans le désordre.
La ligne 14Ø affiche A$, et la ligne 15Ø affiche les numéros de lettres de l'alphabet pour la permutation.
On entre la permutation en ligne 19Ø.
Les lignes 2ØØ et 21Ø transforment la variable alphanumérique entrée en une variable numérique.
La ligne 22Ø appelle une première fois le sous-programme d'inversion.
La ligne 25Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 26Ø comptabilise les essais.
La ligne 17Ø contrôle les résultats.

POUR JOUER :
Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de 1 à 9.
A la question INVERSION (I/J), on répond en entrant un nombre de deux chiffres xy ; x est le numéro de la première lettre, et y le numéro de la seconde. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer un jeu contre l'ordinateur, qui évaluerait le score du joueur en fonction de ses propres performances ; on peut aussi modifier les règles qui régissent les permutations.

```
    10 REM ORDRE
    40 CLS
    50 A$=""
    60 FOR I=1 TO 10
    70   R$=CHR$(RND(10)+64)
    80   FOR J=1 TO I-1
    90          IF R$=MID$(A$,J*2-1,1) TH
EN J=I:NEXTJ:GOTO 70
   100   NEXT J
   110 A$=A$+R$+" "
   120 NEXT I
   130 S=1
   140 PRINT TAB(0,4);"===  RESULTAT : ";
A$
   150 PRINT TAB(16,6);"0 1 2 3 4 5 6 7 8
 9 "
   160 PRINT TAB(0,10);"ESSAI N.";S
   170 IF A$="A B C D E F G H I J " THEN
PRINT TAB(0,22);"VOUS AVEZ TROUVE EN ";S
;" COUPS ...":END
   190 INPUT TAB(0,13);"INVERSION DE I ET
 DE J (TAPEZ IJ) : " R$
   200 I=ASC(LEFT$(R$,1))-48
   210 J=ASC(RIGHT$(R$,1))-48
   220 GOSUB500
   230 IF I=0 OR I=9 THEN 260
   240 I=I-1:J=I+2
   250 GOSUB 500
   260 S=S+1
   270 GOTO 140
   500 B$=""
   510 FOR T=1 TO LEN(A$) STEP 2
   520   IF T=I*2+1 THEN N$=MID$(A$,T,2)
   530   IF T=J*2+1 THEN M$=MID$(A$,T,2)
   540   NEXT T
   550 FOR T=1 TO LEN(A$) STEP 2
   560   IF T=I*2+1 THEN B$=B$+M$:GOTO 59
0
   570   IF T=J*2+1 THEN B$=B$+N$:GOTO 59
0
   580   B$=B$+MID$(A$,T,2)
   590   NEXT T
   595 C$=A$
   600 A$=B$
   610 RETURN
>VDU 12,3
```

Le pendu

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de lettres (version deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affichera alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

LE PROGRAMME :
Le mot à deviner est introduit en ligne 7Ø ; une variable B$, en ligne 2Ø, contient tous les éléments du dessin.
La ligne1ØØ fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
Le sous-programme 27Ø-32Ø s'occupe du dessin (plus ou moins complet, selon la valeur de P).
Les lignes 19Ø-2ØØ affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 12Ø et testée dans une boucle I (14Ø-16Ø) ; le sous-programme 22Ø-24Ø remplace chaque tiret par la lettre correspondante.

POUR JOUER :
Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper RETURN. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question VOTRE LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre sans taper RETURN (GET$).
Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon, la potence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :

Il sera intéressant de constituer une liste de mots dans lesquels Electron pourra piocher au hasard ; on pourra aussi améliorer le dessin (en utilisant VDU 23, par exemple) et modifier le nombre d'essais permis selon la difficulté du mot.

```
   10 REM LE PENDU
   15  DIMV(6),H(6)
   20 B$="### #  ! ##########  - !  ! -
 !  !   !  !  !!  !  !  "
   30 DATA3,8,6,8,6,5,6,11,9,6,9,10
   40 FOR I=1 TO 6: READ V(I),H(I): NEXT
 I
   45  M=0
   50 CLS
   60 COLOUR130: FOR I=1 TO 13:PRINT TAB
(1,I);STRING$(17," "):NEXT I
   70 COLOUR 128:PRINT TAB(20,3);"VOTRE
MOT : ":INPUT TAB(20,5);"====> "R$
   80 L=LEN(R$):IF L>14 OR L=0 THEN PRIN
TCHR$(7);:GOTO70
   90 PRINT TAB(20,3);"MOT DE ";L;" LETT
RES."
  100 P$=STRING$(L,"-")
  110 PRINT TAB(26,5);P$
  120 PRINT TAB(1,17);"TAPER VOTRE LETTR
E : ";:K$=GET$:PRINTK$
  130 T=0
  140 FOR I=1 TO L
  150   IF MID$(R$,I,1)=K$ THEN GOSUB 22
0
  160   NEXT I
  170 IF T=0 THEN M=M+1:GOSUB 270
  180 PRINT TAB(26,5);P$
  190 IF P$=R$ THEN COLOUR 129:PRINT TAB
(0,21);"BRAVO. VOUS AVEZ TROUVE ...":COL
OUR 128:END
  200 IF M=6 THEN COLOUR 129:PRINT TAB(0
,21);"DOMMAGE. VOUS AVEZ PERDU ...":PRIN
T TAB(0,23);"LE MOT ETAIT ";R$:COLOUR 12
8:END
  210 GOTO 120
  220 T=1
  230 P$=LEFT$(P$,I-1)+K$+MID$(P$,I+1)
  240 RETURN
  270 COLOUR 130:COLOUR 1
  280 PRINT TAB(H(M),V(M));MID$(B$,M*9-8
,3)
  290 PRINT TAB(H(M),V(M)+1);MID$(B$,M*9
-5,3)
  300 PRINT TAB(H(M),V(M)+2);MID$(B$,M*9
-2,3)
  310 COLOUR 128:COLOUR 3
  320 RETURN
>VDU 12,3
```

Belote

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté du jeu : facile.

LE JEU :
Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 2Ø cartes sur 32.

LE PROGRAMME :
En ligne 5Ø, on crée un tableau pouvant contenir les 2Ø cartes.
La boucle I (8Ø-14Ø) assure le tirage aléatoire de 2Ø nombres, pris parmi 32 ; la boucle J (11Ø-13Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (16Ø-31Ø) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 21Ø à 28Ø.

POUR JOUER :
Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).
Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :
On pourra facilement mettre au point un affichage plus graphique en utilisant les caractères graphiques redéfinissables de l'Electron ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeux de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```
   10 REM BELOTE
   40 CLS
   50 DIM A(20)
   60 K=1
   70 PRINT TAB(7,12);"===  PATIENTEZ ..
. JE TIRE ..."
   80 FOR I=1 TO 20
   90   A(I)=RND(32)
  100   IF I=1 THEN 140
  110   FOR J=1 TO I-1
  120     IF A(I)=A(J) THEN J=I:NEXT J:G
OTO 90
  130   NEXT J
  140 NEXT I
  150 CLS
  160 FOR I=1 TO 20
  170   S$="CA"
  180   S=(A(I) MOD 8)+1
  190   R$=STR$(S+6)+"   "
  200   R$=LEFT$(R$,2)
  210   IF S=5 THEN R$="VA"
  220   IF S=6 THEN R$="DA"
  230   IF S=7 THEN R$="RO"
  240   IF S=8 THEN R$="AS"
  250   IF A(I)<=24 THEN S$="TR"
  260   IF A(I)<=16 THEN S$="PI"
  270   IF A(I)<=8 THEN S$="CO"
  280   PRINT TAB(10*INT((I-1)/5),2*K);R
$;" ";S$
  290   K=K+1
  300   IF K=6 THEN K=1
  310   NEXT I
>VDU 12,3
```

Un taquin de calcul

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur.
Le codeur part d'un nombre (proposé par Electron) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :
La ligne 5Ø permet d'obtenir l'entier de départ compris entre 5 et 2Ø.
Une boucle I (11Ø-2ØØ) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ et 4ØØØ).
Après chaque calcul, le résultat est affiché (seul le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 19Ø).
La ligne 21Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :
On frappe RUN pour lancer le programme.
Le nombre du début est affiché, puis la question ENTREZ 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de RETURN.
Le résultat sera affiché après chaque opération entrée. A la fin, les opérations entrées sont toutes affichées sur une même ligne, et le message BON COURAGE est envoyé à l'adresse du décodeur (il n'y a pas de contrôle au décodage).

EXTENSIONS POSSIBLES :
On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser Electron participer davantage en lui confiant le rôle du codage, etc.

```
   10 REM UN TAQUIN DE CALCUL
   40 CLS
   50 X=RND(16)+4
   60 PRINT"NOMBRE INITIAL .... : ";X
   70 PRINT:PRINT"ENTRER 5 OPERATIONS"
   80 PRINT"SOUS LA FORME +5 , /2 , *4 .
.."
   90 F=X
  100 DIM R$(5)
  110 FOR I=1 TO 5
  120   PRINT TAB(16,6);"        "
  130   INPUT TAB(0,6);"VOTRE OPERATION
: " R$(I)
  140   A=ASC(LEFT$(R$(I),1))-41
  150   IF A<=0 OR A=3 OR A=5 OR A>6 THE
N 120
  160   IF A>3 THEN A=A/2+1
  170   ON A GOSUB 1000,2000,3000,4000
  180   PRINT TAB(14,10);"         "
  190   PRINT TAB(0,10);"NOMBRE OBTENU :
 ";F
  200   NEXT I
  210 PRINT TAB(0,13)"OPERATIONS : ";R$(
4);"  ";R$(1);"  ";R$(5);"  ";R$(3);"  "
;R$(2)
  220 PRINT TAB(0,16);"===  BON COURAGE
!"
  230 END
 1000 F=F*VAL(MID$(R$(I),2))
 1010 RETURN
 2000 F=F+VAL(MID$(R$(I),2))
 2010 RETURN
 3000 F=F-VAL(MID$(R$(I),2))
 3010 RETURN
 4000 F=F/VAL(MID$(R$(I),2))
 4010 RETURN
>VDU 12,3
```

Pendu inversé

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu sur les mots.
Difficulté du jeu : facile.

LE JEU :
Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est Electron qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :
La ligne 5Ø permet d'enregistrer, dans une variable alphanumérique D$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 6Ø.
La boucle I (9Ø-12Ø) imprime des tirets aux emplacements des lettres à trouver.
La boucle en ligne 14Ø sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée en ligne 15Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 17Ø provoque un nouveau tirage, et la boucle I (21Ø-24Ø) réimprime les tirets et les lettres trouvées.
La ligne 25Ø incrémente le compteur d'essais.
La ligne 26Ø affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de RETURN.
Des tirets sont alors affichés (un par lettre) et Electron propose une première lettre, regarde si elle convient et continue automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par Electron ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```
   10 REM PENDU INVERSE
   40 CLS
   50 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
   60 INPUT TAB(0,3);"ENTREZ VOTRE MOT :
 " R$
   70 C=1:T$=""
   80 CLS
   85 PRINT TAB(1,1);D$
   90 FOR I=1 TO LEN(R$)
  100   PRINT TAB(I,5);"-"
  110   T$=T$+"-"
  120   NEXT I
  130 PRINT TAB(0,9);"JE CHERCHE ...
     ESSAI N.";C
  140 FOR J=1 TO 1000:NEXT J
  150 X=-6*(RND(1)<.7)-13*(RND(1)<.5)-7*
(RND(1)<.4)
  155 IF X=0 THEN 150
  160 X=RND(X)
  170 IF MID$(D$,X,1)="-" THEN 150
  180 E$=MID$(D$,X,1)
  190 D$=LEFT$(D$,X-1)+"-"+MID$(D$,X+1)
  195 PRINT TAB(X,1);"-"
  200 PRINT TAB(0,16);"JE PRENDS : ";E$
  210 FOR I=1 TO LEN(R$)
  220   IF E$=MID$(R$,I,1) THEN T$=LEFT$
(T$,I-1)+E$+MID$(T$,I+1)
  230   PRINT TAB(I,5);MID$(T$,I,1)
  240   NEXT I
  250 C=C+1
  260 IF T$=R$ THEN PRINT TAB(0,18);"TRO
UVE EN ";C-1;" COUPS.":END
  270 GOTO 130
>VDU 12,3
```

Nicomaque

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion (calcul mental).
Difficulté du jeu : moyenne.

LE JEU :
Il utilise une propriété un peu particulière des entiers de 1 à 1ØØ qui peuvent être identifiés sans ambiguīté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :
La ligne 3Ø permet de confectionner l'entier.
La boucle I (7Ø-1ØØ) calcule les trois restes successifs (ligne 8Ø), et permet de les afficher sous une forme assez visuelle, aux bons emplacements (ligne 9Ø).
Le nombre proposé par le joueur est entré à la ligne 11Ø ; il est testé à la ligne 11Ø (affichage de fin de partie : BRAVO ou PERDU).
Les lignes 14Ø et 15Ø s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :
On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des étoiles (l'absence d'étoiles signifiant : reste Ø).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre Ø et 1ØØ, suivi de RETURN.
On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question AUTRE PARTIE (O/N).
Pour faire un nouvel essai, on répondra O, suivi de RETURN.

EXTENSIONS POSSIBLES :
On peut envisager de réaliser le programme décodeur (on donne les restes et Electron devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```
   10 REM NICOMAQUE
   20 CLS
   30 X=RND(100)
   40 PRINT TAB(6,2);"RESTE DANS LA DIVI
SION PAR "
   50 PRINT TAB(0,6)"      3          5
          7"
   60 PRINT TAB(0,8)"      ..         ...
.      ......"
   70 FOR I=1 TO 3
   80   A=X MOD (2*I+1)
   90   PRINT TAB(I*10-4,8);STRING$(A,"*
")
  100   NEXT I
  110 INPUT TAB(0,12);"===  VOTRE ESSAI
: " N
  120 IF N=X THEN PRINT TAB(0,16);"===
BRAVO..."'GOTO140
  130 PRINT TAB(0,16);"===  PERDU ... C'
ETAIT ";X
  140 INPUT TAB(12,20);"UNE AUTRE PARTIE
 ? (O/N)  :  "R$
  150 IF LEFT$(R$,1)="O" THEN20
>VDU 12,3
```

Course de chevaux

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : aucune.

LE JEU :
On entrera un nombre de chevaux compris entre 1 et 9, et la course débutera immédiatement.
Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que Electron s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :
Une première boucle non formelle (8Ø-18Ø) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (1ØØ-2ØØ) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 12Ø fera avancer les chevaux.
La ligne 14Ø gère les arrivées.
La ligne 15Ø réalisera les impressions.
Le sous-programme 24Ø-29Ø fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :
On lance le programme par RUN.
A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 9, suivi de RETURN.
La course se déroulera ensuite automatiquement.

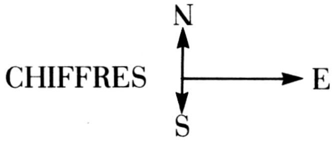
EXTENSIONS POSSIBLES :
On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par leur couleur.
On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

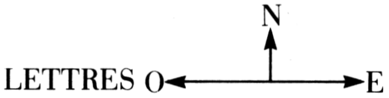
```
   10 REM COURSE DE CHEVAUX
   20 CLS
   30 INPUT TAB(0,1);"NOMBRE DE CHEVAUX
(1 A 9) : " N
   40 IF N<1 OR N>9 THEN 30
   50 A$=""
   60 P=0:A=0:T=1:K=0
   70 J=T
   80 A$=A$+"0":I=J
   90 GOTO 150
  100 FOR I=1 TO N
  110    IF MID$(A$,I,1)="K" THEN 200
  120    B$=CHR$(ASC(MID$(A$,I,1))+RND(4)
)
  130    A$=LEFT$(A$,I-1)+B$+MID$(A$,I+1)
  140    IF B$>="K" THEN GOSUB 240
  150    PRINT TAB(ASC(MID$(A$,I,1))-46,2
*I+3);"     ";I
  160    FOR D=1 TO 200:NEXT D
  170    IF K<>0 THEN200
  180    J=J+1:IF J<=N THEN 80
  190     K=1:GOTO 100
  200    NEXT I
  210 T=0
  220 IF A<N THEN 100
  230 END
  240  A$=LEFT$(A$,I-1)+"K"+MID$(A$,I+1)

  250 P=P+1
  260 PRINT TAB(0,2*I+3);P
  270 T=1
  280 A=A+T
  290 RETURN
>VDU 12,3
```

A travers

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.
L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres. Les lettres peuvent se déplacer vers le haut, à droite et à gauche et les chiffres vers la droite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.





LE PROGRAMME :
Les positions des chiffres et des lettres sont enregistrées dans une table N (4, 4) ; les lignes 3Ø à 7Ø définissent les valeurs de départ.
Les lignes 9Ø à 16Ø dessinent le terrain de jeu et la ligne 17Ø affiche le joueur qui a le trait.
La ligne 2ØØ contrôle le coup joué, de même que les boucles I et J (21Ø à 25Ø), et le changement de joueur est assuré par la ligne 34Ø.

POUR JOUER :
Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné.
On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est) ou O (pour Ouest).
On pourra ainsi faire 2E suivi de RETURN.
Si le coup est invalide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.
Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

EXTENSIONS POSSIBLES :
On peut envisager d'étendre le jeu avec davantage de chiffres ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour Electron.

```
   10 REM A TRAVERS
   20 DIM N(4,4):K=1
   30 FOR I=1 TO 4
   40   N(I,1)=ASC("0")+I
   50   N(4,I)=ASC("?")+I
   60   NEXT I
   70 N(4,1)=0
   80 MODE 5
   90 COLOUR 1:COLOUR 130
  100 FOR I=2 TO 18:PRINT TAB(2,I);STRIN
G$(17," "):NEXT I
  110 COLOUR 1:COLOUR 128
  120 FOR U=1 TO 4
  130   FOR V=1 TO 4
  140     PRINT TAB(4*U,4*V);:IF N(V,U)=
0 THEN PRINT"." ELSE PRINT CHR$(N(V,U))

  150     NEXT V
  160   NEXT U
  170 COLOUR 1:PRINT TAB(0,21);"C'EST AU
X ";:IF K=0 THEN PRINT"LETTRES " ELSE PR
INT "CHIFFRES"
  180 PRINT TAB(0,23);"VOTRE COUP :"
  190 INPUT TAB(6,24);"(N,S,E,O) : " R$
  200 IF (K=1 AND RIGHT$(R$,1)="O") OR (
K=0 AND RIGHT$(R$,1)="S") OR LEN(R$)<>2
THEN 190
  210 FOR I=1 TO 4
  220   FOR J=1 TO 4
  230     IF N(I,J)=ASC(LEFT$(R$,1)) THE
N A=I:B=J:I=4:J=4:NEXT J:NEXT I:GOTO 270
  240 NEXT J
  250 NEXT I
  260 GOTO 170
  270 I=A:J=B
  280 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,
1)="N")
  290 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,
1)="O")
  300 IF (I=0 AND K=0) OR (J>4 AND K=1)
THEN 330
  310 IF N(I,J)<>0 THEN 170
  320 N(I,J)=ASC(LEFT$(R$,1))
  330 N(A,B)=0
  340 K=ABS(K-1)
  350 GOTO 110
>VDU 12,3
```

Mastermind

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 6), en disposant à chaque essai de renseignements.
En effet, Electron répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :
Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 4Ø et 5Ø.
Le nombre est fabriqué à la ligne 6Ø (il pourra y avoir des chiffres répétés).
L'affichage est préparé par les lignes 9Ø et 1ØØ.
La ligne 13Ø s'occupe de la fin de partie.
Une première boucle I (14Ø-23Ø) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (16Ø-18Ø) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 2ØØ et 22Ø.
La ligne 24Ø affiche le résultat, et la ligne 25Ø comptabilise les essais.

POUR JOUER :
On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de RETURN.
Après l'affichage NO. NOMBRE BON PLACE, on introduit son propre essai, suivi de RETURN.
On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter).
On introduit ensuite le nouvel essai, et ainsi de suite.
Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN ... COUPS.
On peut sortir du programme à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats précédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille...).

```
   10 REM MASTERMIND
   20 CLS
   30 E=1:F=1
   40 INPUT TAB(0,2);"A COMBIEN DE CHIFF
RES (<=6) : " N
   50 IF N>6 OR N<1 THEN 40
   60 X$=STR$(RND(10^N)-1)
   70 X$=RIGHT$("000000"+X$,N)
   80 CLS
   90 PRINT TAB(0,1);"N.       NOMBRE
 BON  PLACE"
  100 PRINT TAB(0,2);"-----------------
-----------"
  110 PRINT TAB(0,2+F);E:INPUT TAB(8,2+F
);"" E$
  120 X=0:Y=X
  130 IF E$=X$ THEN PRINT TAB(0,28);"===
 TROUVE EN ";E;" COUPS.":END
  140 FOR I=1 TO N
  150   IF MID$(X$,I,1)=MID$(E$,I,1) THE
N 220
  160   FOR J=1 TO N
  170     IF J<>I AND MID$(X$,I,1)=MID$(
E$,J,1) THEN J=N:NEXT J:GOTO 200
  180   NEXT J
  190 GOTO 230
  200 Y=Y+1
  210 GOTO 230
  220 X=X+1
  230 NEXT I
  240 PRINT TAB(19,2+F);X;TAB(24,2+F);Y
  250 E=E+1:F=F+1:IF F=20 THEN F=1
  260 GOTO 110
>VDU 12,3
```

Idem

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de calcul mental (2 joueurs).
Difficulté du jeu : selon le niveau choisi.

LE JEU :
Electron choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par Electron en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.
Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

LE PROGRAMME :
Une variable numérique N est dimensionnée à 2 en ligne 4Ø, pour contenir les résultats successifs des deux joueurs.
La ligne 5Ø permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 7Ø. La quantité qui interviendra dans les décrémentations est obtenue en ligne 9Ø.
La ligne 12Ø teste si le résultat n'est pas trop petit.
La ligne 13Ø affiche le numéro du joueur.
Le nombre est affiché en ligne 14Ø et le jeu du joueur est entré en ligne 16Ø (nombre de deux chiffres).
La ligne 17Ø incrémente la colonne choisie, et la ligne 18Ø décrémente le nombre de la quantité choisie au départ.
Le résultat sera affiché un court instant (ligne 21Ø). La boucle I (26Ø-28Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 29Ø. Les lignes 3ØØ et 31Ø s'occupent du changement de joueur.

POUR JOUER :
Lancer l'exécution du programme par RUN.
Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant xy, suivi de RETURN ; x est le chiffre à ajouter, et y est le numéro de la colonne où il faut l'ajouter.
Le résultat (provisoire s'il est trop petit) sera alors affiché un court instant, et le trait passe au deuxième joueur, et ainsi de suite. On peut arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On pourra étendre le jeu à plus de 2 joueurs, améliorer le choix du nombre intervenant dans la décrémentation, améliorer les divers contrôles, laisser les différents résultats affichés, etc.

```
   10 REM IDEM
   20 CLS
   30 H=1
   40 DIM N(2)
   50 INPUT TAB(0,1);"QUEL EST LE NIVEAU
 ? (2-7) : " N:IF N<2 OR N>7 THEN 50
   60 U=10:P=0
   70 N(1)=INT((U^N-U^(N-1))*RND(1)+U^(N
-1))
   80 N(2)=N(1)
   90 A=INT(2*U^(N-1)*RND(1)+U^(N-1))
  100 IF H=1 THEN P=P+1
  110 CLS
  120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(
N(H))
  130 PRINT TAB(0,3);"JOUEUR N.";H;"
 ESSAI N.";P
  140  PRINT TAB(0,6);"NOMBRE : ":PRINT
TAB(16-N,6);N(H)
  150 PRINT TAB(9,7);"7654321 "
  160 INPUT TAB(0,10);"===  VOTRE JEU (S
OUS LA FORME NC) : "R$:IF LEN(R$)<>2 THE
N 160
  170 N(H)=N(H)+VAL(LEFT$(R$,1))*U^(VAL(
RIGHT$(R$,1))-1)
  180 N(H)=N(H)-A
  190 IF N(H)>U^N THEN N(H)=N(H)-U^N
  200 N(H)=INT(N(H))
  210 PRINT TAB(0,13);"===  RESULTAT = "
;N(H)
  220 PRINTCHR$(7);
  230 FOR J=1 TO 1200:NEXT J
  240 R$=STR$(N(H))
  250 R$=MID$(R$,2)
  260 FOR I=1 TO LEN(R$)-1
  270     IF MID$(R$,I,1)<>MID$(R$,I+1,1)
 THEN I=LEN(R$):NEXT I:GOTO 300
  280 NEXT I
  290 PRINT TAB(0,20);"=== BRAVO, JOUEUR
 N.";H:END
  300 H=H+1
  310 IF H>=3 THEN H=1
  320 GOTO 100
>VDU 12,3
```

Horloge 1

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

LE PROGRAMME :
La ligne 3Ø vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1Ø24 signifera 1Ø heures 24 minutes).
Les lignes 8Ø à 19Ø réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 18Ø).
Les lignes 2ØØ à 23Ø assurent la progression de l'heure.
Les lignes 25Ø à 3ØØ effectuent l'affichage de l'heure et de deux points clignotants pour les secondes.

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape et on attend le top pour appuyer sur RETURN.
L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche ESCAPE.
L'heure sera entrée sous forme d'un entier ayant impérativement 4 chiffres, pour 5 heures 6 minutes, on veillera à taper Ø5Ø6 suivi de RETURN.

EXTENSIONS POSSIBLES :
Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

```
   10 REM HORLOGE 1
   20 CLS
   30 INPUT TAB(0,2);"HEURE (FORME HHMM)
 : " A$
   40 IF LEN(A$)<>4 THEN 30
   50 T=TIME:T1=T
   60 COLOUR 129:CLS:COLOUR 130:COLOUR 1
   70 K=1
   80 A=VAL(MID$(A$,K,1))
   90 IF A<>1 AND A<>4 THEN PRINT TAB(6*
K-5,1);"    "
  100 FOR J=1 TO 5
  110   IF A=0 OR (A>3 AND A<7) THENPRI
NT TAB(6*K-5,J);" "
  120   IF A<>5 AND A<>6 THEN PRINT TAB(
6*K-3,J);" "
  130   IF A=0 OR A=2 OR A=6 OR A=8 THEN
 PRINT TAB(6*K-5,J+4);" "
  140   IF A<>2 THEN PRINT TAB(6*K-3,J+4
);" "
  150   NEXT J
  160 IF A>1 THEN PRINT TAB(6*K-4,5);"
"
  170 IF A<>1 AND A<>4 AND A<>7 THEN PRI
NT TAB(6*K-5,9);"    "
  180 K=K+1
  190 IF K<5 THEN 80
  200 A=1+VAL(A$)
  210 IF A=2360 THEN A=0
  220 IF (A MOD 100)=60THEN A=A+40
  230 A$=RIGHT$("000"+STR$(A),4)
  240 H=1
  250 IF TIME-T>=6000THENT=TIME:GOTO60
  260 IF TIME-T1<50 THEN 260
  270 T1=TIME
  280 PRINT TAB(24,9);RIGHT$("0"+STR$(IN
T((TIME-T)/100)),2)
  290 PRINT TAB(11,9);:IF H=1 THEN PRINT
":" ELSE PRINT" "
  300 H=ABS(H-1):GOTO250
>VDU 12,3
```

Reines sur l'échiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Le programme utilise un algorithme très performant pour résoudre un problème qui a donné du fil à retordre à des chercheurs, même aussi célèbres que le mathématicien GAUSS, qui n'avait pas pu découvrir toutes les solutions. Il s'agit de trouver toutes les manières de disposer des reines sur un échiquier, de telle sorte qu'aucune ne puisse en prendre une autre. Le programme proposé est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme visuelle et sous forme numérique.

LE PROGRAMME :
La ligne 3Ø vous permet de choisir la taille de l'échiquier.
Un tableau D sera dimensionné en conséquence, en ligne 4Ø.
Une boucle d'exploration principale I (5Ø-11Ø) contient une sous-boucle J (8Ø-1ØØ) qui réalise le contrôle de non-alignement (ligne 9Ø).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 23Ø à 27Ø et on recommence.
La boucle F (13Ø-21Ø) assure l'impression de l'échiquier, et la boucle K (16Ø-18Ø), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 22Ø permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :
On lance le programme avec RUN, puis on tape la dimension de l'échiquier désirée, et on fait RETURN.
Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue). La première solution sera affichée puis le programme s'arrête, et il faut appuyer sur la touche RETURN pour le faire repartir. Pour terminer l'exploration sans avoir eu toutes les solutions, il faudra taper ESCAPE.

EXTENSIONS POSSIBLES :
On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.).

```
   10 REM REINES SUR L'ECHIQUIER
   20 CLS
   30 INPUT TAB(0,2);"===  TAILLE DE L'E
CHIQUIER : " S
   40 DIM D(S)
   50 I=1
   60 D(I)=1
   70 IF I=1 THEN 110
   80 FOR J=1 TO I-1
   90   IF D(I)=D(J) OR (ABS(D(I)-D(J))=
I-J) THEN J=I:NEXTJ:GOTO240
  100 NEXT J
  110 I=I+1:IF I<=S THEN 60
  120 CLS
  130 FOR F=1 TO S
  140   PRINT TAB(2*F+1,0);F
  150   PRINT TAB(0,2*F);F
  160   FOR K=1 TO S
  170     PRINT TAB(2*K+1,2*F);"."
  180     NEXT K
  190   PRINT TAB(2*D(F)+1,2*F);"*"
  200   PRINT TAB(F,20);RIGHT$(STR$(D(F)
),1)
  210   NEXT F
  220 INPUT TAB(0,25);"TAPER 'RETURN' PO
UR CONTINUER : "R$
  230 I=I-1
  240 D(I)=D(I)+1
  250 IF D(I)<=S THEN 70
  260 I=I-1
  270 IF I<>0 THEN 240
>VDU 12,3
```

Répétitions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Des nombres vont apparaître un par un.
Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il en aura un de plus à taper à chaque fois).
Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

LE PROGRAMME :
La ligne 6Ø appelle un sous-programme (lignes 1ØØØ à 1Ø6Ø) qui donne la règle du jeu.
La ligne 8Ø fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$.
La ligne 9Ø permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 1ØØ.
La ligne 12Ø attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 13Ø.
La ligne 16Ø marque les points.
La ligne 17Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 2ØØ à la ligne 24Ø.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE.
On lui répond en redonnant le chiffre, suivi de RETURN. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface, et de nouveau la question VOTRE SEQUENCE. On retape alors les deux chiffres déjà obtenus, suivis de RETURN.
On continue ainsi (3 chiffres, puis 4, 5, etc.) jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O/N) ; on tape O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant ESCAPE avant l'affichage d'un chiffre.

EXTENSIONS POSSIBLES :
On pourra réaliser des jeux analogues et plus ou moins perfectionnés en remplaçant les chiffres par des lettres ou des couleurs ou des sons...

```
   10 REM REPETITIONS
   40 M=0
   50 A$=""
   60 GOSUB 1000
   70 H=0
   80 X=RND(10)-1:A$=A$+STR$(X)
   90 CLS:PRINT TAB(RND(40)-1,4);X
  100 FOR I=1 TO 200+RND(300) : NEXT I
  110 CLS
  120 INPUT TAB(0,10)"== VOTRE SEQUENCE
: " R$
  130 IF R$<>A$ THEN 190
  140 PRINT TAB(0,14);"== EXACT ==   ATT
ENTION ..."
  150 FOR I=1TO 400 : NEXT I
  160 H=H+1
  170 IF M<H THEN M=H
  180 GOTO 80
  190 CLS
  200 PRINT TAB(0,2);"VOTRE SCORE EST DE
 ";H
  210 PRINT TAB(0,5);"MEILLEUR SCORE ACT
UEL : ";M
  220 INPUT TAB(0,8);"AUTRE PARTIE ? (O/
N) : " R$
  230 IF R$="N" THEN END
  240 GOTO 50
 1000 CLS
 1010 PRINT TAB(0,3);"VOUS DEVEZ REPETER
 TOUTE LA SERIE DES"
 1020 PRINT TAB(0,5);"NOMBRES APPARUS DE
PUIS LE DEBUT DU JEU"
 1030 PRINT TAB(0,8);"IL Y EN A UN DE PL
US A CHAQUE FOIS ..."
 1040 INPUT TAB(0,12);"TAPER 'RETURN' PO
UR COMMENCER : " R$
 1050 RETURN
 1060 RETURN
>VDU 12,3
```

Mots

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de lettres (pour 1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il est difficile d'imaginer plus simple dans son principe que ce jeu, qui se contente d'afficher dans le désordre les lettres prises dans une trentaine de mots. Mais essayez de jouer, et vous verrez que ce n'est pas si facile, la difficulté variant évidemment selon la liste de mots que vous aurez introduits en DATA au préalable.

LE PROGRAMME :
La ligne 20 permet de dimensionner la variable A$ (I) qui contiendra les mots.
La boucle I (30-50) saisit la liste des mots (inscrits en DATA aux lignes 240 à 280).
La ligne 70 choisit un mot au hasard ; la boucle I (100-150) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.
La ligne 160 affiche les lettres, et la ligne 170 attend votre proposition, qui est testée en ligne 180.
Le nombre d'essais est incrémenté en ligne 190.
La fin de partie est contrôlée par la ligne 200, et le score est affiché en ligne 220.

POUR JOUER :
On lance le programme par RUN. Une série de lettres est affichée dans le désordre, et il faut essayer de retrouver le mot correspondant.
On tape le mot supposé, puis RETURN, et une nouvelle série de lettres sera affichée, si l'on répond O à la question AUTRE ESSAI (O/N).
Le score et le nombre d'essais sont affichés en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer de minuter le temps d'apparition du mot, ou de faire défiler les lettres l'une après l'autre sur l'écran.
On peut aussi corser la difficulté en introduisant des JOKERS, jouer à plusieurs, tenir le compte des points, etc.

```
   10 REM MOTS
   20 DIM A$(30):S=0:E=0
   30 FOR I=1 TO 30
   40   READ A$(I)
   50   NEXT I
   60 CLS
   70 Z=RND(30)
   80 IF A$(Z)="" THEN 70
   90 R$=A$(Z):A$="":A$(Z)="":V$=R$
  100 FOR I=1 TO LEN(R$)
  110   Z=RND(LEN(R$))
  120   IF MID$(R$,Z,1)="-" THEN 110
  130   A$=A$+MID$(R$,Z,1)
  140   R$=LEFT$(R$,Z-1)+"-"+MID$(R$,Z+1
)
  150   NEXT I
  160 PRINT TAB(0,3);"LES LETTRES SONT :
 ";A$
  170 INPUT TAB(0,6);"VOTRE SOLUTION   :
 " S$
  180 IF S$=V$ THEN PRINT TAB(35,7);"O.K
.":S=S+1
  190 E=E+1
  200 INPUT TAB(0,10);"===  AUTRE ESSAI
? (O/N) : " R$
  210 IF LEFT$(R$,1)="O" THEN 60
  220 PRINT TAB(0,15);"===  VOTRE SCORE
: ";S;"/";E
  230 END
  240 DATAJAMBON,ENERGUMENE,AFFILIE,ERUD
IT,INSIGNIFIANT,REALISTE
  250 DATAIMPRESSIONNANT,ALLECHANT,EPIQU
E,INTREPIDE,COUARD,ORDINATEUR
  260 DATATREVE,BANDEROLLE,SENAT,ACTUEL,
CONSACRE,DIEDRE
  270 DATALIVRE,TERMINE,SIGNATURE,MALODO
RANT,AIMANT,ARBITRAIRE
  280 DATACHALEUR,OPPORTUN,PELURE,ALTRUI
STE,RAMONEUR,NOVICE
>VDU 12,3
```

Permutations

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Electron vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).
Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :
La ligne 30 permet de choisir le nombre de lettres, et la ligne 40 contrôle que l'on n'en demande pas plus de 12.
La ligne 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 100), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 110-120).
La ligne 140 contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 160, 170 et 180.
Le retournement est fait, lettre par lettre, dans la boucle I (210-230) : le résultat est placé dans B$ en ligne 240, affiché par la ligne 250 et comparé au résultat final en ligne 260.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 270.

POUR JOUER :
Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) : on répond en frappant N, suivi de RETURN.
L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à obtenir (FIN).
On joue en indiquant le nombre de lettres à permuter à partir de la gauche, par exemple : sur un affichage initial de GEABDFC, si l'on tape 3 et RETURN, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ainsi qu'une question AUTRE PARTIE (O/N) : répondre O si l'on désire faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra utiliser le même type de jeu pour, par exemple, reconstituer un dessin caché dont les diverses parties seraient repérées par des lettres. On peut aussi réaliser des permutations plus complexes (par exemple le fait de permuter deux lettres permute les deux voisines de la lettre la plus à gauche, etc.).

```
   10 REM PERMUTATIONS
   20 CLS
   30 INPUT TAB(0,2);"NOMBRE DE LETTRES
(DE 1 A 12) : " N
   40 IF N<1 OR N>12 THEN 30
   50 A$="ABCDEFGHIJKL"
   60 B$="":C$="            "
   70 CLS
   80 X=RND(N)
   90 IF MID$(A$,X,1)=" " THEN 80
  100 B$=MID$(A$,X,1)+B$
  110 C$=LEFT$(C$,X-1)+MID$(A$,X,1)+MID$
(C$,X+1)
  120 A$=LEFT$(A$,X-1)+" "+MID$(A$,X+1)
  130 S=1
  140 IF LEN(B$)<>N THEN 80
  150 C$=LEFT$(C$,LEN(B$))
  160 PRINT TAB(0,4);"DEBUT ..... : ";B$
  170 PRINT TAB(0,7);"FIN ....... : ";C$
  180 PRINT TAB(0,11);"PERMUTATION NUMER
O ";S:INPUT TAB(0,13);"==  VOTRE QUANTIT
E : "L:IF L>N THEN 180
  190 S=S+1
  200 A$=""
  210 FOR I=1 TO L
  220   A$=A$+MID$(B$,L-I+1,1)
  230   NEXT I
  240 B$=A$+MID$(B$,L+1)
  250 PRINT TAB(0,17);"==  RESULTAT : ";
B$
  260 IF B$<>C$ THEN 180
  270 PRINT TAB(0,19);"    TROUVE EN ";S
-1;" ESSAIS."
  280 INPUT TAB(0,22);"AUTRE PARTIE ? (O
/N) : " R$
  290 IF LEFT$(R$,1)="O" THEN20
>VDU 12,3
```

Bridge

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème posé.

LE JEU :
Vous ne trouverez pas, dans ce programme, un partenaire pour vos bridges mais, plus exactement, une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu. On peut, à partir de là, se poser toutes sortes de problèmes : annonces, jeu de la carte, etc.

LE PROGRAMME :
La ligne 50 définit un jeu de 52 cartes.
La variable B$, en ligne 70, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).
La boucle J (80-220) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (90-190) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 100.
La ligne 110 calcule la couleur de la carte tirée.
La ligne 120 vérifie que cette carte n'a pas déjà été tirée.
Les lignes 130 et 140 s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 150, 160 et 170.
La ligne 180 permet d'éliminer les cartes déjà tirées, et la ligne 200 réalise un arrêt entre chaque affichage.

POUR JOUER :
On lance le programme par RUN.
Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur la touche RETURN pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :
On peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.
On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles de l'Electron, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour Electron.

```
   10 REM BRIDGE
   40 CLS
   50 R$=STRING$(4,"234567891VDRA")
   70 B$="TRCACOPIOAAOS"
   80 FOR J=1 TO 4
   90   FOR I=1 TO 13
  100     X=RND(LEN(R$))
  110     S=2*INT((X-1)/13)+1
  120     IF MID$(R$,X,1)="O" THEN 100
  130     T=X MOD 13
  140     T=T-13*(T=0)
  150     PRINT TAB(10*J-10,2*I);MID$(R$
,X,1);
  160     IF T>=9 THENPRINT MID$(B$,T,1)
  170     PRINT TAB(10*J-6,2*I);MID$(B$,
S,2)
  180     R$=LEFT$(R$,X-1)+"O"+MID$(R$,X
+1)
  190     NEXT I
  200   INPUT TAB(0,29);"TAPER 'RETURN'
POUR CONTINUER : "S$
  210   PRINT TAB(0,29);STRING$(40," ")
  220   NEXT J
>VDU 12,3
```

Tours de Hanoï

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (8 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque de diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :
La variable D$, en ligne 4Ø, servira à dessiner les disques.
Le nombre de disques est entré en ligne 5Ø et refusé s'il y a plus de 8 disques.
La boucle I (7Ø-9Ø) permet de fabriquer une variable A (1) qui contient l'état de la première pile sous forme numérique (en ligne 3Ø, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 11Ø et les trois piles par les boucles emboîtées I (14Ø-19Ø) et J (15Ø-18Ø).
Le contrôle du coup est effectué par la ligne 33Ø et l'état des piles est modifié par les lignes 34Ø et 35Ø.

POUR JOUER :
Après un RUN qui lance le programme, on entre le nombre de disques demandé, suivi de RETURN. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis à la question TOUR D'ARRIVEE on répond par le numéro de la tour d'arrivée. Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :
On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.
On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques de couleur différente, par exemple.

```
   10 REM TOURS DE HANOI
   20 CLS:COLOUR 3
   30 DIMA(3):A(1)=9:A(2)=9:A(3)=9
   40 C$=STRING$(9,"="):D$=STRING$(9,"#"
)
   50 INPUT TAB(0,2);"NOMBRE D'ANNEAUX (
1 A 8) : " N
   60 IF N<1 OR N>8 THEN 50
   70 FOR I=0 TO N-1
   80   A(1)=VAL(STR$(A(1))+STR$(N-I))
   90   NEXT I
  100 CLS
  110 PRINT TAB(0,14);C$;TAB(12);C$;TAB(
24);C$
  120 COLOUR 2
  130 PRINT TAB(4,14);1:PRINTTAB(16,14);
2:PRINTTAB(28,14);3
  140 COLOUR 1:FOR I=0 TO N-1
  150   FOR J=1 TO 3
  160     B$=STR$(A(J))
  170      PRINT TAB(12*J-11,13-I);:IF L
EN(B$)>I+1 THEN PRINT LEFT$(D$,VAL(MID$(
B$,I+2,1))) ELSE PRINT STRING$(9," ")
  180     NEXT J
  190   NEXT I
  200 COLOUR 3
  210 PRINT TAB(0,18);"TOUR DE DEPART :
     TOUR D'ARRIVEE : "
  220 PRINT TAB(17,18);:D=VAL(GET$)
  230 IF D<1 OR D>3 THEN 220
  240 IF A(D)=9 THEN 220
  250 PRINT TAB(17,18);D
  260 PRINT TAB(39,18);:A=VAL(GET$)
  270 IF A<1 OR A>3 THEN 260
  280 B$=STR$(A(D))
  290 A$=STR$(A(A))
  300 GOTO330
  310 B$=MID$(STR$(A(D)),2)
  320 A$=MID$(STR$(A(A)),2)
  330 IF RIGHT$(B$,1)>RIGHT$(A$,1) THEN
220
  340 A(A)=VAL(A$+RIGHT$(B$,1))
  350 A(D)=VAL(LEFT$(B$,LEN(B$)-1))
  360 GOTO140
>VDU 12,3
```

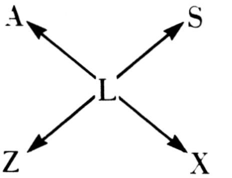
Le loup et les agneaux

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 1Ø sur 1Ø. Il est présenté, ici, dans une version pour un damier de 8 cases : l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut.
Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut : les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :
Une variable A$, en ligne 4Ø contient les positions de départ.
La boucle I (8Ø-15Ø) dessine le damier, et les lignes 16Ø à 21Ø placent le loup et les agneaux.
La ligne 22Ø gère le changement de joueur.
La ligne 25Ø affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 24Ø JEU DES AGNEAUX dans le cas contraire.
Les lignes 26Ø à 33Ø analysent la réponse, en fonction du joueur concerné.

POUR JOUER :
RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et RETURN si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (A, S, Z, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant A, S, Z ou X, suivant la direction choisie :



On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :
On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...
On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant à Electron de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour Electron).

```
   10 REM LE LOUP ET LES AGNEAUX
   20 COLOUR128:COLOUR3:CLS
   30 INPUT TAB(0,2);"AGNEAUX (0) OU LOU
P (1) : " S:IF S<0 OR S>1 THEN 30
   40 A$="8183858714"
   50 CLS
   60 PRINT"   A  B  C  D  E  F  G  H"
   70 FOR I=1 TO 8:PRINT TAB(0,3*I);I:NE
XT I
   80 FOR I=1 TO 8
   90   FOR J=1 TO 8
  100     IF ((I+J) MOD 2)=0 THEN COLOUR
 129 ELSE COLOUR 130
  110     FOR K=-1 TO 1
  120       PRINT TAB(3*J-1,3*I+K);"   "
  130       NEXT K
  140      NEXT J
  150    NEXT I
  160 COLOUR 130:COLOUR 0
  170 PRINT TAB(3*VAL(MID$(A$,10,1)),3*V
AL(MID$(A$,9,1)));"L"
  180 PRINT TAB(3*VAL(MID$(A$,2,1)),3*VA
L(MID$(A$,1,1)));"1"
  190 PRINT TAB(3*VAL(MID$(A$,4,1)),3*VA
L(MID$(A$,3,1)));"2"
  200 PRINT TAB(3*VAL(MID$(A$,6,1)),3*VA
L(MID$(A$,5,1)));"3"
  210 PRINT TAB(3*VAL(MID$(A$,8,1)),3*VA
L(MID$(A$,7,1)));"4"
  220 S=ABS(S-1)
  230 PRINT TAB(0,28);STRING$(40," ")
  240 IF S=1 THEN INPUT TAB(0,28);"JEU D
ES AGNEAUX (ND OU NG) : " R$:GOTO 310
  250 INPUT TAB(0,28);"JEU DU LOUP (A,S,
Z,X) : " R$
```

```
  260 C=10
  270 PRINT TAB(3*VAL(MID$(A$,10,1)),3*V
AL(MID$(A$,9,1)));" "
  280 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X
")+(C-1 AND R$="Z")
  290 A=A-(C+1 AND R$="A")-(C-1 AND R$="
S")
  300 GOTO 340
  310 C=2*VAL(R$)
  320 PRINT TAB(3*VAL(MID$(A$,C,1)),3*VA
L(MID$(A$,C-1,1)));" "
  330 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT
$(R$,1)="D")-(11 AND RIGHT$(R$,1)="G")
  340 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)
+RIGHT$(A$,LEN(A$)-C)
  350 GOTO 160
>VDU 12,3
```

Niveau 4

Nom du programme	*Thèmes et particularités*
Losanges	Pour vous aider à vous endormir.
Mathieu	Vous ne savez pas dessiner ? Laissez faire l'ordinateur.
Anagrammes	Un bon jeu de réflexe : pour deux joueurs
Tirage de dés	Suffisamment souple pour être utilisé partout.
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Sous-marin	Un classique des jeux vidéo (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ?
Décomaque	Pour vous éviter de sécher trop longtemps sur "Nicomaque".
Clavier	Apprenez à taper à la machine en vous amusant.
Clavier Bis	Maintenant, entraînez-vous à taper vite !

Les techniques d'animation élémentaire

SAISIE AVEC ATTENTE D'UN CARACTÈRE GET$ ET GET

Ces instructions permettent de saisir un caractère sans que l'utilisateur ait besoin de taper RETURN.

GET$ renvoit le caractère et GET son code ASCII.

Ainsi : K = GET provoquera l'attente d'une frappe de touche, et K contiendra le code ASCII de la touche frappé.

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : INKE Y$ et INKEY

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic Electron, cette propriété est offerte, par la **fonction INKEY$** qui effectue une scrutation des touches du clavier et enregistre le nom de la dernière touche frappée, ou la fonction INKEY qui enregistre son code ASCII.

On pourra utiliser INKEY$ de deux façons :

- soit dans une boucle (réalisation d'un mouvement). On saisit INKEY$ et on le teste pour une éventuelle sortie de la boucle.
- soit dans une attente : tant que INKEY$ est vide (on n'a rien frappé) on attend, et l'on sort dès qu'une touche a été frappée.

L'argument de la fonction, s'il est positif, détermine le *temps d'attente* de la frappe.

Il sera quelquefois nécessaire de taper plusieurs fois la touche pour qu'elle soit prise en compte.

DEPLACEMENTS D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SCROLL

Il est difficile, si l'on travaille en Basic, de gérer le déplacement simultané de plusieurs objets sur l'écran (sinon on obtient vite une animation saccadée, à cause d'un temps de déplacement prohibitif).

Electron réalise automatiquement un **"Scrolling"** (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran (son numéro dépend du mode utilisé).

Cette propriété a été mise à profit dans plusieurs jeux : cela permet en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est bien évidemment perdue.

Si un objet doit rester immobile sur l'écran, il sera nécessaire de l'effacer dans sa nouvelle position, et de le replacer à son emplacement initial.

Losanges

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.

LE JEU :
Le programme va progressivement construire sur l'écran un losange, puis il va en modifier les couleurs, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.

LE PROGRAMME :
Une première boucle K (5Ø-14Ø) définit les dimensions des tracés successifs.
La deuxième boucle I (7Ø-13Ø) trace des bandes de points, en utilisant des instructions PLOT (lignes 8Ø à 12Ø).
Si la variable S (ligne 8Ø) est à 1, alors il y aura modification de la couleur.

POUR JOUER :
On commence l'exécution du programme avec RUN et on regarde.
On peut arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On pourra introduire le son, et imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).

```
   10 REM LOSANGES
   20 MODE 2
   30  COLOUR 128:CLS
   40 S=-1
   50 FOR P=20 TO 1 STEP -1
   60   FOR K=0 TO 300 STEP P
   70     FOR I=0 TO K STEP P
   80       IF S<>0 THEN ON RND(5) GOTO
190,200,210,220,230
   90       PLOT70,640-K+I,512-I
  100       PLOT70,640-K+I,512+I
  110       PLOT70,640+K-I,512+I
  120       PLOT70,640+K-I,512-I
  130       NEXT I
  140     NEXT K
  150    S=NOT S
  170   NEXT P
  180 END
  190 GCOL 0,1:PLOT 69,640-K+I,512-I
  200 GCOL 0,2:PLOT 69,640-K+I,512+I
  210 GCOL 0,3:PLOT 69,640+K-I,512+I
  220 GCOL 0,4:PLOT 69,640+K-I,512-I
  230 GOTO 130
>VDU 12,3
```

Mathieu

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.

LE JEU :
On utilise la définition d'une courbe en coordonnées polaires et la fonction PLOT pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.

LE PROGRAMME :
Les 3 premiers paramètres Z, K et T, introduits en ligne 2Ø, définissent le rayon maximum et l'occupation de l'écran.
Les variables aléatoires A et B, choisies entre 1 et 4, de la ligne 6Ø, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 6Ø INPUT A, B).
La boucle H (7Ø-12Ø) réalise le tracé, point par point. Le rayon R est calculé en ligne 8Ø, et le passage en coordonnées cartésiennes est effectué aux lignes 9Ø et 1ØØ.
L'affichage des points est réalisé en ligne 11Ø.
La ligne 13Ø imprime un message de fin.
La ligne 15Ø permet de refaire un nouveau dessin, alors que les lignes 16Ø et 17Ø assurent un nouveau tracé sans effacement du précédent.

POUR JOUER :
On lance le programme en tapant RUN.
Le dessin sera ensuite exécuté automatiquement.
Le message "c'est fini" signale la fin du tracé.
Si on veut un nouveau tracé, on tape Ø.
Si on veut un tracé sans effacement du dessin précédent, on tape une touche quelconque.
Pour arrêter, faire ESCAPE.

EXTENSIONS POSSIBLES :
On pourra essayer d'utiliser d'autres fonctions pour réaliser des tracés encore plus extraordinaires (par exemple avec simulation de la troisième dimension).
Les deux paramètres principaux A et B pourront être, si nécessaire, entrés à la main (instructions INPUT) pour faire une étude plus systématique des différentes courbes possibles.

```
  10 REM MATHIEU
  20  Z=20:T=5:K=10
  30 MODE 2
  40 C=RND(7)
  50 GCOL 0,C
  60  A=RND(4):B=RND(4)
  70 FOR H=0 TO 6.28 STEP 3.14/180
  80   R=Z*SIN(H*T)
  90   X=640+INT(K*R*COS(A*H))
 100   Y=512+INT(K*R*SIN(B*H))
 110   PLOT 69,X,Y
 120   NEXT H
 130 PRINT TAB(0,27);"---  C'EST FINI .
.."
 140 A$=GET$
 150  IF A$="O" THEN 30
 160 C=RND(7)
 170 GOTO 50
>VDU 12,3
```

Anagrammes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (2 joueurs).
Difficulté du jeu : selon le niveau.

LE JEU :
Des lettres sont choisies au hasard par Electron, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou Ø) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :
Le niveau du jeu est entré à la ligne 5Ø. La boucle I (8Ø-1ØØ) réalise le choix des lettres.
Une deuxième boucle I (12Ø-19Ø) effectue le brouillage, et la ligne 14Ø permet de détecter la pression d'une touche.
La ligne 2ØØ affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
Les lignes 21Ø-22Ø gèrent le tableau des mots affichés.
La boucle I (27Ø-29Ø) explore les mots présents sur l'écran, le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (lignes 3ØØ-31Ø).

POUR JOUER :
On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire RETURN. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou Ø), et son résultat sera affiché après un court instant.
La partie sera alors terminée.

EXTENSIONS POSSIBLES :
On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

```
   10 REM ANAGRAMMES
   20 DIM A$(23)
   30 FOR I=0 TO 23:A$(I)="":NEXT
   40 MODE 6
   50 INPUT TAB(0,0);"NIVEAU (DE 1 A 3)
: " N
   60 IF N<1 OR N>3 THEN 50
   70 S$=""
   80 FOR I=1 TO N+3
   90   S$=S$+CHR$(RND(26)+64)
  100   NEXTI
  110 A$="":B$=S$
  120 FOR I=1 TO N+3
  130   K=RND(N+3)
  140   D=INKEY(0)-48
  150   IF D=0 OR D=1 THEN 260
  160   IF MID$(B$,K,1)="-" THEN 130
  170   A$=A$+MID$(B$,K,1)
  180   B$=LEFT$(B$,K-1)+"-"+MID$(B$,K+1
)
  190   NEXT I
  200 PRINT TAB(0,24);A$
  210 FORI=0TO22:A$(I)=A$(I+1):NEXT
  220 A$(23)=A$
  230 FOR I=1 TO 20*(N+3) : NEXT I
  240 S$=A$
  250 GOTO 110
  260 K=-1
  270 FOR I=0 TO 22
  280   IF S$=A$(I) THEN K=I
  290   NEXT I
  300 IF K<>-1 THEN PRINT TAB(10,K);"<--
- BRAVO JOUEUR DE"; ELSE PRINT TAB(10,23
);"DOMMAGE JOUEUR DE ";
  310 IF D=0 THEN PRINT" DROITE" ELSE PR
INT" GAUCHE"
  320 SOUND 1,2,100,8
  330 END
>VDU12,3
```

Tirage de dés

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.
Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :
La ligne 6Ø saisit le caractère frappé au clavier et le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).
Une variable D dimensionnée à 9 reçoit la valeur indiquée par chaque dé.
Une boucle J (12Ø-14Ø) est utilisée pour dessiner un fond, et une autre boucle I (11Ø-17Ø) sert au tirage (tirage en ligne 15Ø et impression en ligne 16Ø).
La ligne 18Ø offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :
On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.
Le résultat est alors automatiquement affiché ; et le programme repart.
Pour arrêter, il est nécessaire de taper Ø en nombre de dés.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```
   10 REM TIRAGE DE DES
   20 DIM D(9)
   40 CLS
   50 PRINT TAB(0,2);"NOMBRE DE DES A LA
NCER (1 A 9) : ";
   60 A$=GET$
   65 IF A$="0" THEN 190
   70 IF A$<"1" OR A$>"9" THEN 60
   80 PRINTA$
   90 N=VAL(A$)
  100 CLS
  110 FOR I=1 TO N
  115   COLOUR 130:COLOUR 1
  120   FOR J=1 TO 3
  130     PRINT TAB(4*I-4,8+J);"    "
  140     NEXT J
  150   D(I)=RND(6)
  160   PRINT TAB(4*I-3,10);D(I)
  170   NEXT I
  180 COLOUR128:COLOUR3:GOTO50
  190 PRINT TAB(0,22)
>VDU 12,3
```

Jeu de loto

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents, qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

LE PROGRAMME :
On utilisera un tableau N(7), dimensionné en ligne 3Ø, pour enregistrer les résultats.
Le tirage des nombres est effectué dans une boucle I (6Ø-12Ø), une deuxième boucle J (9Ø-11Ø) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.
Le tirage (ou le retirage) est effectué en ligne 7Ø.
La boucle I (19Ø-21Ø) affiche le résultat, et la ligne 23Ø affiche le numéro complémentaire.

POUR JOUER :
Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.

EXTENSIONS POSSIBLES :
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.
On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le Midi de la France.

```
   10 REM JEU DE LOTO
   20 MODE 2
   30 DIM N(7)
   40 COLOUR 128
   50 CLS
   60 FOR I=1 TO 7
   70   N(I)=RND(49)
   80   IF I=1 THEN 120
   90   FOR J=1 TO I-1
  100     IF N(I)=N(J) THEN J=I:NEXT J:G
OTO 70
  110   NEXT J
  120 NEXT I
  130 COLOUR 130:COLOUR 1
  140 PRINT TAB(2,1);STRING$(16," ")
  150 PRINT TAB(2,2);" TIRAGE DU LOTO "
  160 PRINT TAB(2,3);" ";STRING$(14,"_")
;" "
  170 PRINT TAB(2,4);STRING$(16," ")
  180 PRINT TAB(0,11);STRING$(60," ")
  190 FOR I=1 TO 6
  200   PRINT TAB(3*I-2,12);N(I)
  210   NEXT I
  220 PRINT TAB(0,19);STRING$(60," ")
  230 PRINT TAB(0,20);" N. COMPLEM. : ";
N(7)
  240 COLOUR 128:COLOUR 7:PRINT TAB(0,22
)
>VDU 12,3
```

La bombe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes censé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (1ØØ*1ØØ*1ØØ).
Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 1Ø en 1Ø.
Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :
Une variable numérique R, dimensionnée à 3, contient les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 5Ø.
Les boucles I (8Ø et 9Ø-11Ø) dessinent le détecteur.
Le signal de proximité est calculé en lignes 12Ø-13Ø et la ligne 14Ø permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 15Ø permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 16Ø réalise le décompte du temps.
La ligne 17Ø contrôle si l'on a trouvé ou non.
La boucle I (21Ø-25Ø) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 27Ø à 31Ø.

POUR JOUER :
Après le lancement du programme par RUN, le signal et le temps (en principe 2ØØ) sont affichés.
Le joueur est alors en position x=Ø, y=Ø, z=Ø, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.
On fera, par exemple, 13 puis RETURN, 45 puis RETURN, etc.
Le nouveau signal est alors affiché, et le temps est décrémenté de 1Ø.
On continuera ainsi, soit jusqu'au bon résultat (arrêt du jeu et fin de programme), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

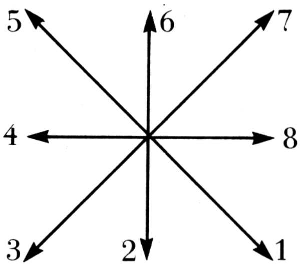
EXTENSIONS POSSIBLES :
Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

```
   10 REM LA BOMBE
   20 DIM R(3)
   30 L=100
   40 T=200
   50 A=RND(L)-1:B=RND(L)-1:C=RND(L-1)
   60 COLOUR 128:CLS
   70 COLOUR 129
   80 FOR I=2 TO 8:PRINT TAB(10,I);STRIN
G$(16," "):NEXT I
   90 COLOUR 130:FOR I=4 TO 6
  100    PRINT TAB(11,I);"        ":PRINT
 TAB(20,I);"      "
  110    NEXT I
  120 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3
)*L))
  130  S$=LEFT$(STR$(S),7):S=VAL(S$)
  140 IF T=0 THEN PRINT TAB(0,25);"===
PERDU ... ":PRINT TAB(0,27);"===  LA SOL
UTION ETAIT : ";A;",  ";B;",` ";C:END
  150 COLOUR 0:PRINT TAB(12,5);S:PRINT T
AB(21,5);T;" "
  160 T=T-INT(SQR(L))
  170 IF S=0 THEN PRINT TAB(0,25);"===
BRAVO.":END
  180 PRINT TAB(10,12);"- X : "
  190 PRINT TAB(10,14);"- Y : "
  200 PRINT TAB(10,16);"- Z : "
  210 FOR I=1 TO 3
  220    R$=""
  230    GOSUB 270
  240    R(I)=VAL(R$)
  250    NEXT I
  260 GOTO 90
  270 COLOUR 128:PRINT TAB(16,I*2+10);"
  ":COLOUR 130:PRINT TAB(16,I*2+10);
  280 A$=GET$
  290 IF ASC(A$)=13 THEN RETURN
  300 PRINT A$;:R$=R$+A$
  310 GOTO 280
>VDU 12,3
```

Ardoise magique

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 8, suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche Ø (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur Ø.
On peut choisir la couleur du point en appuyant sur une touche de A à H ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME :
La ligne 3Ø contient le choix des possibilités de couleur.
La ligne 5Ø imprime le point élémentaire.
Les lignes 6Ø à 9Ø contrôlent les entrées.
La ligne 13Ø appelle l'un des quatre sous-programmes effectuant le déplacement.
La ligne 14Ø vérifie que l'on ne sort pas des limites de l'écran.
La ligne 16Ø effectue l'effacement s'il y a lieu.
Le sous-programme 4ØØ-41Ø choisit la couleur ; 3ØØ-31Ø réalise ou non l'effacement.

POUR JOUER :
On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper Ø pour le déplacer sans afficher.
Si on frappe une lettre de A à D, la couleur du point change.
La touche P efface tout, et l'on recommence.
On peut sortir à tout moment, en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, et travailler en mode 16 couleurs, etc.

```
   10 REM ARDOISE MAGIQUE
   15 S=0
   20 MODE 1
   30 C$="0123"
   40 X=640:Y=512:C=3
   50 GCOL 0,C:PLOT 69,X,Y
   60 A$=GET$
   70 IF A$>="A" AND A$<="D" THEN GOSUB
400
   80 IF A$="O" THEN GOSUB 300
   90 IF A$="P" THEN GOTO 20
  100 F=-2*(VAL(A$)<5)-1
  110 A=X:B=Y
  120 U=VAL(A$)+2*(F-1)
  121  F=4*F
  125 IF U=0 THEN 140
  130 ON U GOSUB 1000,2000,3000,4000
  140 D=-(X>0 AND X<1279 AND Y>0 AND Y<1
023)
  150 IF D=0 THEN GOSUB 5000
  160 IF S<>0 THEN PLOT 71,A,B
  170 GOTO 50
  300 S=ABS(S-1)
  310 RETURN
  400 C=VAL(MID$(C$,ASC(A$)-64,1))
  410 RETURN
 1000 Y=Y+F
 2000 X=X+F
 2010 RETURN
 3000 GOSUB 2000
 4000 Y=Y-F
 4010 RETURN
 5000 X=A:Y=B
 5010 RETURN
>VDU 12,3
```

Affichage publicitaire

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (4 ou 5 lignes) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi. Surtout, ne mettez pas de virgules dans votre texte !

LE PROGRAMME :
Le texte sera introduit à la ligne 5Ø, puis formatté à plus de 8Ø caractères.
Le défilement vers la gauche ou vers la droite est préparé à la ligne 14Ø.
Le texte sera affiché en ligne 16Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 17Ø (durée de l'attente de la frappe d'une touche).
La ligne 18Ø permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :
On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RETURN.
On obtient ensuite l'affichage DEFILEMENT A (**D**)ROITE OU (**G**)AUCHE auquel on répondra en tapant, soit D, soit G.
Puis à la question VITESSE, on répondra en introduisant un grand nombre pour une vitesse rapide, et un petit pour une vitesse lente (par exemple, de Ø à 4Ø ou plus).
Le texte commencera à défiler dès que l'on aura frappé RETURN.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur une touche quelconque pour retrouver les questions SENS et VITESSE.
On tapera ESCAPE pour interrompre le programme.

EXTENSIONS POSSIBLES :
On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```
   10 REM AFFICHAGE PUBLICITAIRE
   20 CLS
   30 B$=""
   40 PRINT TAB(0,1);"ENTREZ VOTRE TEXTE
 " B$
   50 INPUTTAB(0,3);""B$
   60  C$=B$+" "
   70 IF LEN(C$)<80 THEN C$=C$+B$+" ":GO
TO 70
   80 PRINT TAB(0,6);"DEFILEMENT (D)ROIT
E OU (G)AUCHE : ";
   90 S$=GET$:IF S$<>"D" AND S$<>"G" THE
N 90
  100 IF S$="D" THEN PRINT"DROITE" ELSE
PRINT"GAUCHE"
  110 INPUT TAB(0,9);"VITESSE (DE 0 A 50
) : "V:IF V<0 OR V>50 THEN 110
  120 V=51-V
  130 CLS
  140 IF S$="G" THEN D=0:F=LEN(C$)/2:P=1
 ELSE D=LEN(C$)/2:F=0:P=-1
  150 FOR I=D TO F STEP P
  160   PRINT TAB(0,18);MID$(C$,I+1,40)
  170   R=INKEY(V)
  180   IF R<>-1 THEN I=F:NEXT I:GOTO 80
  190 NEXT I
  200 GOTO 150
>VDU 12,3
```

Evasion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Votre seul problème : réussir à traverser l'écran, sans que l'Electron vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :
La ligne 7Ø définit la position de départ du joueur.
La ligne 8Ø choisit la coordonnée horizontale de la case où Electron va tirer (liée à votre position), et la ligne 9Ø choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 18Ø et 19Ø ; sa trajectoire est tracée par la ligne 15Ø, et sa position actuelle (sur l'écran, un #) est imprimée par la ligne 2ØØ.
La ligne 21Ø regarde s'il est touché ou non.
La ligne 22Ø effectue le contrôle de fin de partie.

POUR JOUER :
On commence l'exécution du programme par RUN.
Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un #.
Les 4 touches de direction (←, ↓, ↑, →) servent à déplacer le joueur.
S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.
S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.
On peut terminer à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```
   10 REM EVASION
   20 ENVELOPE 3,129,-14,-7,-2,10,10,10,
0,0,0,0,0,0
   30 MODE 1
   40 *FX4,1
   50 GCOL 0,2
   60 COLOUR 1
   70 X=0:Y=12
   80 H=32*X+RND(64)-16
   90 K=RND(1024)-1
  100 IF X<3 THEN 150
  110 FOR J=1 TO RND(5)+2
  120   PLOT 69,H+RND(30)-15,K+RND(30)-1
5
  130   SOUND &11,3,250,2
  140   NEXT J
  150 PRINT TAB(X,Y);"."
  160 A=INKEY(1)
  170 *FX15,1
  180 X=X-(A=137)+(A=136)
  190 Y=Y-(A=138)+(A=139)
  200 PRINT TAB(X,Y);"#"
  210 IF X>3 AND (ABS(H-(32*X+16))+ABS(K
-(1024-(32*Y+16))) <= 34) THEN PRINT TAB
(0,29);"=== DOMMAGE. VOUS AVEZ PERDU ...
":GOTO 240
  220 IF X<38 THEN 80
  230 PRINT TAB(0,29);"=== BRAVO. VOUS A
VEZ GAGNE."
  240 *FX4,0
>VDU 12,3
```

Combat aérien

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.
La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :
La ligne 6Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 8Ø et 9Ø.
Les lignes 12Ø et 13Ø vous permettent le pilotage de votre appareil.
La ligne 17Ø dessine le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 2ØØ à la ligne 25Ø et le test de fin de partie est effectué en ligne 24Ø.
La ligne 18Ø regarde si vous avez tiré ou non.

POUR JOUER :
Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche Ø.
L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.
La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.
On peut aussi arrêter à tout moment en frappant ESCAPE.

EXTENSIONS POSSIBLES :
Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 18Ø).
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```
   10 REM COMBAT AERIEN
   20 *FX4,1
   30 CLS
   40 A=5:B=15
   50 N=28
   60 PRINT TAB(B,A);"#=#"
   70 U=A:V=B
   80 A=A+2*RND(2)-3
   90      B=B+2*RND(2)-3
  100 K=INKEY(0)
  110 *FX15,1
  120 B=B-2*(K=137)+2*(K=136)
  130 A=A-2*(K=138)+2*(K=139)
  140 IF B<0 OR B>37 THEN B=ABS(B-37)
  150 IF A<0 OR A>28 THEN A=ABS(A-28)
  160 PRINT TAB(V,U);"    "
  170 PRINT TAB(18,12);"* *":PRINT TAB(1
8,13);" X ":PRINT TAB(18,14);"* *"
  180 IF K=48 OR N<28 THEN GOSUB 200
  190 GOTO 60
  200 PRINT TAB(32-N,N);"/":PRINT TAB(N+
6,N);"\"
  210 PRINT TAB(31-N,N+1);" ":PRINT TAB(
N+7,N+1);" "
  220 N=N-1
  230 IF N>12 THEN RETURN
  240 IF (A=12 OR A=13 OR A=14) AND (B=1
8 OR B=19 OR B=20) THEN PRINT TAB(0,28);
"=== BRAVO. VOUS AVEZ GAGNE...":GOTO 26
0
  250 GOTO 50
  260 *FX4,0
>VDU 12,3
```

Dessin de dés

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 5 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :
La ligne 6Ø contrôle la fin du jeu, et la ligne 7Ø la validité de la réponse.
Les lignes 9Ø et 1ØØ initialisent les deux variables de dessin des dés.
La boucle I (11Ø-2ØØ) et les lignes 12Ø à 135 dessinent le cadre du dé.
La ligne 14Ø réalise le tirage, et la boucle J (15Ø-19Ø) dessine les points en relation avec le tirage réalisé (en appelant les variables A$ et B$).

POUR JOUER :
On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée.
Il suffira de taper Ø pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toute sorte de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```
   10 REM DESSIN DE DES
   20 CLS
   30 PRINT TAB(0,17);"NOMBRE DE DES (1
A 5) : ";
   40 A$=GET$
   50 N=VAL(A$)
   60 IF N=0 THEN 220
   70 IF N<1 OR N>5 THEN 40
   80 CLS
   90 A$="       #  #        ##   #"
  100 B$="121314324515525555"
  110 FOR I=1 TO N
  120   PRINT TAB(8*I-7,1);"-----"
  125   PRINT TAB(8*I-7,9);"-----"
  130   PRINT TAB(8*I-8,2) ;"!     !"
  135   PRINT TAB(8*I-8,8) ;"!     !"
  140   X=RND(6)
  150   FOR J=1 TO 5
  160     IF J<=3 THEN S=5*VAL(MID$(B$,3
*(X-1)+J,1))-4
  170     IF J<=3 THEN PRINT TAB(8*I-7,2
*J+1);MID$(A$,S,5)
  180     PRINT TAB(8*I-8,J+2);"!":PRINT
 TAB(8*I-2,J+2);"!"
  190     NEXTJ
  200   NEXTI
  210 GOTO 30
  220 PRINT
>VDU 12,3
```

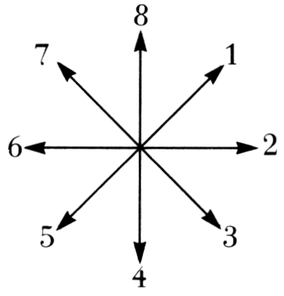
Le jeu du missile

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : contre Electron.
Difficulté du jeu : moyenne.

LE JEU :
Electron est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie et de revenir sain et sauf sur votre base ; mais, dès que l'Electron vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire.
Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :
La ligne 4Ø dessine la base ennemie à survoler (sous forme d'un H agrandi).
La ligne 5Ø dessine votre base de départ.
Les lignes 8Ø et 1ØØ choisissent l'instant de départ du missile.
La ligne 12Ø dessine votre avion.
Les lignes 14Ø et 15Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1ØØØ, 2ØØØ, 3ØØØ et 4ØØØ.
La ligne 16Ø assure le départ du missile qui est dessiné par la ligne 17Ø et dont le déplacement est géré par les lignes 18Ø et 19Ø.
Les lignes 2ØØ à 23Ø contrôlent la fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).
Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper ESCAPE pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...). On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
   10 REM LE JEU DU MISSILE
   20 MODE1:CLS
   30 M=0
   40 PRINT TAB(17,4);"##";TAB(17,5);"##
"
   50 COLOUR129:PRINT TAB(0,19);"    ";T
AB(0,20);"     "
   60 COLOUR 128:COLOUR 2
   70 U=15:X=18
   80 S=RND(U)+5
   90 Y=1
  100 S=S-1
  110 V=33
  120 PRINT TAB(Y,X);"Y"
  130 A$=GET$
  140 F=-2*(VAL(A$)<5)-1
  150 ON VAL(A$)+2*(F-1) GOSUB 1000,2000
,3000,4000
  160 IF S<>0 THEN 100
  170 PRINT TAB(V,U);"<"
  180 U=U-SGN(U-X)
  190 V=V-SGN(V-X)
  200 IF U=X AND V=Y THEN PRINT TAB(0,25
);"===  TOUCHE...":END
  210 IF (X=4 OR X=5) AND (Y=17 OR Y=18)
 THEN 20
  220 IF X<=4 AND Y>=18 THEN M=1
  230 IF M=1 AND X=19 AND Y<=4 THEN PRIN
T TAB(0,25);"===  MISSION ACCOMPLIE.":EN
D
  240 GOTO 120
 1000 X=X-F
 2000 Y=Y+F
 2010 RETURN
 3000 GOSUB 2000
 4000 X=X+F
 4010 RETURN
>
```

Le singe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.
Le rôle du singe est, ici, tenu par Electron qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.
Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou Ø).
Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.
Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME :
La ligne 3Ø contient une liste V$ des voyelles courantes et la ligne 4Ø une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc.).
La ligne 5Ø définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (8Ø-13Ø) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 1ØØ), soit dans la liste V$ (ligne 11Ø).
La ligne 16Ø temporise l'apparition du résultat en attendant la frappe d'une touche pendant 2,5 secondes.
La ligne 17Ø gère la fin de partie, avec affichage du score.
La ligne 18Ø regarde si l'un des joueurs a réagi.
Enfin, les lignes 2ØØ et 21Ø comptabilisent les points de chaque joueur.

POUR JOUER :
On lance le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper Ø.
Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.
Si l'un des joueurs a frappé Ø ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de RETURN.
Les points sont alors calculés par Electron, et la série de lettres suivante est alors affichée.

Pour sortir du programme, il faudra taper P au lieu de Ø ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :
On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
   10 REM LE SINGE
   20 A=0:B=0
   30 V$="AEIOU"
   40 C$="BCDGLMNPRSTV"
   50 X=RND(4)+2
   60 M$=""
   70 CLS
   80 FOR I=1 TO X
   90   E$=CHR$(RND(26)+64)
  100   IF RND(1)<.4 THEN E$=MID$(C$,RND
(LEN(C$)),1)
  110   IF RND(1)<.5 THEN E$=MID$(V$,RND
(LEN(V$)),1)
  120   M$=M$+E$
  130   NEXT I
  140 PRINT TAB(5,5);CHR$(7);M$;CHR$(7)
  160 A$=INKEY$(250)
  170 IF A$="P" THEN PRINT TAB(0,20);"SC
ORE DE A : ";A;"     SCORE DE B : ";B:EN
D
  180 IF A$<>"0" AND A$<>"1" THEN 50
  190 INPUT TAB(5,9);"(J)USTE OU (F)AUX
: " R$
  200 A=A-(R$="J" AND A$="0")+(R$="F" AN
D A$="0")
  210 B=B-(R$="J" AND A$="1")+(R$="F" AN
D A$="1")
  220 GOTO 50
>VDU 12,3
```

Le jeu du 15

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.
Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :
La boucle I (6Ø-8Ø) dessine le terrain de jeu, et les lignes 9Ø à 1ØØ s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 11Ø attend le coup joué.
La boucle I (12Ø-14Ø) vérifie la validité du coup.
La ligne 15Ø teste la fin de partie, en cas de match nul, et la ligne 16Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
Les lignes 17Ø et 18Ø s'occupent de l'affichage du coup, et la ligne 19Ø l'enregistre dans une variable R$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 21Ø à la ligne 3ØØ : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 31Ø.

POUR JOUER :
Lancer le jeu en tapant RUN.
Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.
On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.
En cas de match nul, la machine affiche C'EST FINI.
On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :
Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).
On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```
10 REM LE JEU DU 15
20  MODE1:COLOUR 131:CLS
30 F=0:R$="123456789"
40 O$="":X$=""
50 COLOUR 129 : FOR I=2 TO 6:PRINT TA
B(1,I);STRING$(37," "):NEXT
60 FOR I=1 TO 9:FOR J=3 TO 5
70     COLOUR 128:PRINT TAB(I*4-2,J);
"   "
80     NEXT J:COLOUR 131:COLOUR 0:PRI
NT TAB(I*4-1,8);I:NEXT I
90 COLOUR 131:COLOUR 0:PRINT TAB(0,12
);"JOUEUR N.";F+1;
100 IF F=1 THEN PRINT" : O" ELSE PRINT
" : X"
110 A$=INKEY$(0) : IF A$<"1" OR A$>"9"
 THEN 110
120 FOR I=1 TO LEN(R$)
130   IF A$=MID$(R$,I,1) THEN GOSUB 17
0
140   NEXT I
150 IF VAL(R$)=0 THEN PRINT TAB(0,20);
"----- C'EST FINI -----":END
160 GOTO 90
170 IF F=1 THEN O$=O$+A$:COLOUR 128:CO
LOUR 2:PRINT TAB(I*4-1,4);"O"
180 IF F=0 THEN X$=X$+A$:COLOUR 128:CO
LOUR 2:PRINT TAB(I*4-1,4);"X"
190 R$=LEFT$(R$,I-1)+"0"+MID$(R$,I+1)
200 IF F=1 THEN T$=O$ ELSE T$=X$
210 IF LEN(T$)<3 THEN 310
220 FOR I=1 TO 3
230   IF I=LEN(T$) THEN I=3:GOTO 300
240   FOR J=I+1 TO LEN(T$)
250     IF J=LEN(T$) THEN I=3:GOTO290
260     FOR K=J+1 TO LEN(T$)
270       IF VAL(MID$(T$,I,1))+VAL(MID
$(T$,J,1))+VAL(MID$(T$,K,1))=15 THEN 330
280       NEXT K
290     NEXT J
300   NEXT I
310 F=ABS(F-1)
320 RETURN
330 PRINT TAB(0,12);"BRAVO, JOUEUR ";
340 IF F=1 THEN PRINT"O" ELSE PRINT "X
"
>
```

Adresse

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Des chiffres, entre Ø et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.
Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 1 point. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :
Les lignes 5Ø et 6Ø contiennent, dans deux variables alphanumériques C$ et B$, les références qui seront utilisées pour dessiner les chiffres en grand format.
L'emplacement du chiffre tiré par Electron est choisi aux lignes 7Ø et 8Ø, et le chiffre est tiré en ligne 9Ø.
La ligne 1ØØ réalise l'impression, et la ligne 11Ø détermine la durée d'affichage. La ligne 13Ø définit la durée du jeu, et la fin de partie est traitée par la ligne 15Ø.
La boucle I (2ØØ-23Ø) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 24Ø.

POUR JOUER :
Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.
Si le résultat est exact, le joueur marquera 1 point.
Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.
En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.
On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```
   10 REM ADRESSE
   20 CLS
   30 T=100
   40 S=0:R=0
   50 C$="#####      ##  #"
   60 B$="1444133333131211313144133121311
21411333314141114131"
   70 X=RND(15)
   80 Y=RND(15)+15
   90 N=RND(10)-1
  100 R=R+1:PRINT TAB(Y,X);N
  110  A=INKEY(RND(100)+100)
  120 IF A<>-1 THEN 190
  130 T=T-N
  140 PRINT TAB(Y,X);"   "
  150 IF T>=0 THEN 70
  160 CLS
  170 PRINT TAB(0,20);"VOTRE SCORE EST D
E ";S;" POINTS SUR ";R
  180 END
  190 U=A-48
  200 FOR I=1 TO 5
  210   A=4*VAL(MID$(B$,5*U+I,1))-3
  220   PRINT TAB(2,2*I);MID$(C$,A,4)
  230   NEXT I
  240 IF U=N THEN S=S+1
  250 GOTO 130
>VDU 12,3
```

Sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : facile.

LE JEU :
Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :
La ligne 7Ø permet à Electron de choisir la profondeur du sous-marin.
La ligne 9Ø dessine votre bâtiment.
Les lignes 1ØØ et 11Ø affichent le score et dessinent la surface de l'eau.
La ligne 12Ø dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 2ØØ à 24Ø.
Le déplacement (très rapide) de votre navire est réalisé par les lignes 17Ø et 18Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 14Ø.

POUR JOUER :
Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux touches 1 et 2.
La touche Ø autorise le tir.
Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.
Pour terminer la partie, il conviendra de faire ESCAPE ou d'attendre 9 sous-marins.

EXTENSIONS POSSIBLES :
Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```
   10 REM SOUS-MARIN
   20 P=0
   30 C=15
   40 S=5
   50 COLOUR 128:CLS:COLOUR 130:PRINTSTR
ING$(160," "):COLOUR 1
   60 COLOUR 128:PRINT TAB(0,S);"
"
   70 T=27:S=RND(13)+5:P=P+.01
   80 X=1
   85  COLOUR 1:COLOUR 130
   90 PRINT TAB(C,1);"  ### ###  ";TAB(C
,2);"   \###/   "
  100 PRINT TAB(0,4);"=================
======================"
  110 COLOUR 128:COLOUR 2:PRINT TAB(0,24
);"SCORE : ";P
  120 PRINT TAB(T,S);" \==/ "
  130 IF X=1 THEN Y=C+2*X
  140 IF S=X AND Y>T AND Y<T+6 THEN P=P+
1 :PRINT TAB(T,S);"*****":FOR I=1 TO 10:
SOUND 1,-1,70-I*2,1:NEXT I
  150 T=T-1:IF P>9 THEN PRINT TAB(0,26);
"C'EST FINI ...":END
  160 IF T<0 THEN 60
  170 A=INKEY(0)-48
  180 C=C-2*((A=2 AND C<27)-(A=1 AND C>1
))
  190 IF A<>0 AND X=1 THEN 85
  200 PRINT TAB(Y,X);:IF X<4 THEN COLOUR
 130 ELSE COLOUR 128
  210 PRINT" ":X=X+1
  220 IF X>S THEN 80
  225 SOUND 1,-1,150,2
  230 COLOUR 1:PRINT TAB(Y,X);"O"
  240  GOTO 85
>VDU 12,3
```

Diago

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : difficile.

LE JEU :
On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 3 secondes. Le jeu commence lorsque l'on arrive à Ø seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche RETURN pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran.
On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :
Le nombre de points noirs sera choisi au hasard, en ligne 3Ø.
Le temps est décompté par les lignes 6Ø à 1ØØ.
Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 12Ø.
La pression de la touche RETURN est détectée en ligne 15Ø, et la ligne 16Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 18Ø à 21Ø.

POUR JOUER :
On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme : IL VOUS RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de Ø, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche RETURN.
L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.
On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

```
   10 REM DIAGO
   20 CLS
   30 X=RND(18)+2
   40 S=3
   50 PRINT TAB(0,1);"===  ATTENTION :":
PRINT"===  LE BON REFLEXE EST A ";X
   60 PRINT TAB(0,10);"IL VOUS RESTE ";S
;" SECONDES ..."
   70 S=S-1
   80 SOUND 1,-1,150,1:FOR I=1 TO 600:NE
XT I
   90 PRINT TAB(14,10);S
  100 IF S<>0 THEN 70
  110 U=0
  120 PRINT TAB(U,31);"O"
  130 U=U+1
  140 A=INKEY(20)
  150 IF A=13 THEN 180
  160 IF U>X THEN PRINT :PRINT"PERDU ...
":GOTO 200
  170 GOTO 120
  180 IF U<>X THEN PRINT :PRINT"PERDU ..
.":GOTO 200
  190 PRINT:PRINT"BRAVO ..."
  200 PRINT:PRINT"AUTRE ESSAI ? (O/N) :
";
  210 A$=GET$ : IF A$="O" THEN 20
>VDU 12,3
```

Décomaque

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solution du programme Nicomaque.
Difficulté du jeu : aucune.

LE JEU :
C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 1ØØ, et Electron calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :
Les lignes 4Ø à 6Ø affichent quelques explications.
La ligne 11Ø attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.
Cette ligne 11Ø emmagasine les restes dans une variable A (S).
La ligne 12Ø dessine des étoiles selon les restes introduits.
Les lignes 18Ø à 21Ø gèrent la fin de partie.

POUR JOUER :
On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DANS LA DIVISION PAR 3" s'affiche.
On doit répondre Ø, 1 ou 2, et l'on obtient le message "RESTE DANS LA DIVISION PAR 5" etc.
La solution est affichée dès que l'on a donné les 3 restes.
Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).
Tapez N pour finir.

EXTENSIONS POSSIBLES :
On peut étendre le programme aux entiers de 1 à 1ØØØ, en utilisant les restes par 3, 5, 7 et 11 ; on pourra rajouter du son et de la couleur pour rendre le programme plus attrayant.

```
   10 REM DECOMAQUE
   20 DIMA(3)
   30 COLOUR 128:CLS
   40 PRINT TAB(0,1);"VOUS DEVEZ ENTRER
LES TROIS RESTES"
   50 PRINT:PRINT"DANS LES DIVISIONS PAR
 3,5 ET 7"
   60 PRINT:PRINT"D'UN NOMBRE ENTIER INF
ERIEUR A 100"
   70 I=3:S=1
   80 COLOUR 1:COLOUR 130
   90 PRINT TAB(0,15);"RESTE DIVISIONS P
AR 3    PAR 5    PAR 7":PRINT TAB(19,17)
;"..";TAB(26,17);"....";TAB(33,17);"....
.."
  100 PRINT TAB(0,10);"RESTE DANS LA DIV
ISION PAR ";I;" ";
  110 A(S)=GET-48:IF A(S)<0 OR A(S)>=I T
HEN 110
  120 PRINT TAB(S*7+12,17);STRING$(A(S),
"*")
  130 S=S+1:I=I+2
  140 IF I<8 THEN 100
  150 PRINT TAB(0,10);STRING$(39," ")
  160 T=A(1)*70+A(2)*21+A(3)*15
  170 T=T MOD 105
  180 PRINT TAB(0,20);"LA SOLUTION EST "
;T
  190 PRINT TAB(0,23);"VOULEZ-VOUS UN AU
TRE ESSAI ? (O/N) : ";
  200 K$=GET$:IF K$="O" THEN 30
  210 COLOUR 128:COLOUR 3:PRINT
>VDU 12,3
```

Clavier

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe (apprentissage du clavier).
Difficulté du jeu : difficile.

LE JEU :
Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; le caractère qu'il convient de frapper au clavier apparaît en inversé. Plus l'on frappe vite, et plus les déplacements s'effectuent rapidement (le temps écoulé depuis le début s'affiche en continu).

LE PROGRAMME :
Deux tableaux X et Y, dimensionnés à 26, contiennent les positions de chacune des lettres de l'alphabet (ligne, colonne). Vous pouvez modifier la disposition du clavier en modifiant les DATA en 6Ø (nombre de touches par ligne) et en 7Ø (ordre des touches).
Le sous-programme (35Ø-38Ø) affiche une touche.
La ligne 19Ø choisit la lettre.
La ligne 27Ø teste votre frappe.

POUR JOUER :
On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui apparaît en inversé, puis une autre touche est affichée.
La durée de la partie n'est pas limitée ; il suffit de taper RETURN pour obtenir l'affichage du score, et du temps total.
Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la colonne par une balle. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```
   10 REM CLAVIER
   20 COLOUR 128:COLOUR 3
   30 CLS
   40 S=0
   50 DIM N(3),X(26),Y(26)
   60 DATA 10,19,26
   70 DATA Q,W,E,R,T,Y,U,I,O,P,A,S,D,F,G
,H,J,K,L,Z,X,C,V,B,N,M
   80 FOR I=1 TO 3:READ N(I):NEXT
   90 COLOUR 1:COLOUR 130
  100 FOR I=1 TO 26:READ K$:K=ASC(K$)-64
  110   FOR J=3 TO 1 STEP -1:IF I<=N(J)
THEN X(K)=J
  120      NEXT J
  130   Y(K)=I-N(X(K)-1)
  140   GOSUB 350
  150   NEXT I
  160 PRINT TAB(0,18);" 0 ESSAIS JUSTES
SUR 0"
  170 PRINT TAB(0,20);"VOTRE TEMPS EST D
E 0"
  180 TIME=0
  190 K=RND(26)
  200 COLOUR 2:COLOUR 129
  210 GOSUB 350
  220 COLOUR 130:COLOUR 1
  230 FOR F=1 TO 40+RND(40)
  240   L=INKEY(0)
  250   IF L=-1 THEN 290
  260   IF L=47 THEN END
  270   IF L=K+64 THEN S=S+1
  280   F=80
  290   PRINT TAB(19,20);INT(TIME/100);"
 S."
  300   NEXT F
  310 P=P+1:PRINT TAB(0,18);S;" ";TAB(21
,18);P;" "
  320 GOSUB 350
  330 GOTO 190
  340 END
  350 PRINT TAB(Y(K)*4-4+X(K),X(K)*4-3);
"   "
  360 PRINT TAB(Y(K)*4-4+X(K),X(K)*4-2);
" ";CHR$(K+64);" "
  370 PRINT TAB(Y(K)*4-4+X(K),X(K)*4-1);
"   "
  380 RETURN
>VDU 12,3
```

Clavier bis

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe.
Difficulté du jeu : suivant votre niveau.

LE JEU :
Il s'agit d'un test de votre vitesse de frappe sur un clavier. Vous tapez un texte de la longueur que vous désirez et l'ordinateur vous indiquera votre vitesse en mots/minute (sachant que les mots français ont en moyenne six caractères, 7 avec l'espace).

LE PROGRAMME :
La ligne 7Ø initialise l'horloge de l'Electron à Ø.
La ligne 11Ø permet de saisir le caractère frappé.
Le temps est affiché en continu par la ligne 12Ø.
Si vous frappez **RETURN** (testé en ligne 15Ø), les lignes 22Ø à 28Ø gèrent l'affichage de votre vitesse.
Vous pouvez alors soit continuer, soit taper un deuxième **RETURN** pour arrêter.

POUR JOUER :
On lance le programme par **RUN.** Attention, le chronomètre démarre aussitôt. A vous de frapper le plus vite possible. A chaque **RETURN,** vous revenez à la ligne et votre résultat intermédiaire est affiché. Deux **RETURN** à la suite vous permettent d'arrêter.

EXTENSIONS POSSIBLES :
On peut faire afficher un texte à recopier, ou dessiner un clavier en indiquant à chaque frappe la touche concernée (voir programme précédent).

```
   10 REM CLAVIER BIS
   20 COLOUR 128:COLOUR 3
   30 MODE1:CLS
   40 S=0
   50 PRINT TAB(0,27);"TEMPS :      NB.CA
R :      NB.MOTS :        "
   60 PRINT TAB(0,29);"VITESSE :"
   70 TIME=0
   80  PRINT TAB(0,1);"TAPER VOTRE TEXTE
 :"
   90 NBC=0:NB2=0
  100 PRINT TAB(0,4);
  110 K$=INKEY$(0)
  120 PRINT TAB(8,27);INT(TIME/100)
  130 IF LEN(K$)=0 THEN 100
  140 T=TIME
  150  IF ASC(K$)=13 THEN 220
  160 AG=0
  170 J=INT(NB2/40)+4:I=NB2 MOD 40
  180 PRINT TAB(I,J);K$
  190 NBC=NBC+1
  200 NB2=NB2+1
  210 GOTO 100
  220 IF AG=1 THEN 290
  230 AG=1
  240 NB2=40*(INT(NB2/40)+1)
  250  NBM=INT(NBC/7)
  260 V=INT(NBM/T*6000)
  270 PRINT TAB(21,27);NBC;TAB(35,27);NB
M;TAB(10,29);V
  280 GOTO 100
  290 END
>
```

Niveau 5

Nom du programme	*Thèmes et particularités*
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs).
Horloge 2	Utilisation du crayon optique de l'ordinateur.
Dames	Pour deux joueurs : classique.
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique dans une version très complète.
Echecs	Un programme assez simplifié, pour deux joueurs.
Calculatrice	Un programme de simulation très réaliste.
Echiquier	Une base graphique pour vos propres créations.
Zombies	Un peu farfelu, mais très visuel (1 joueur).

Le lézard d'Alice	Beaucoup de hasard (et de merveilles).
Chapeaux	Soyez observateur, et vous réussirez (2 joueurs).
Othello	Pour deux joueurs ; il retourne les pions à votre place.

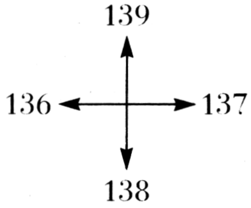
Les techniques spécifiques de l'Electron

INTRUCTIONS ET COMMANDES PARTICULIERES : VDU et *FX

– **VDU x** est équivalent à **PRINT CHR$(x)**

Nous utiliserons ici :
VDU 23, n, liste de huit valeurs 224 ≤ n < 255
permet de définir le caractère graphique de code n.
VDU 28, X1, Y2, X2, Y1
définit une fenêtre texte de coordonnées extrêmes (X1, Y1) et (X2, Y2)
VDU 26
remet la fenêtre texte normale
VDU 19, C1, C2, Ø, Ø, Ø
affecte aux points de couleur n° C1 la couleur C2
VDU 2Ø
remet les couleurs normales

– Les commandes *FXi vous permettent de paramétrer votre Electron.
*FX4,1 autorise la récupération d'un code ASCII par GET ou INKEY en cas de frappe d'une touche de déplacement de curseur :



DEFINITION DE PROCEDURES ET DE FONCTIONS DEFPROC ( ) et DEF FN ( )

Une procédure peut être considérée comme un sous-programme avec passage de paramètre. Ainsi, voici une procédure permettant de définir une couleur de texte et une couleur de fond :

*FX4, Ø supprime cette possibilité.

```
DEF PROCCOL (T,F)
COLOUR T
COLOUR F+128
END PROC
```

L'avantage de la définition de fonction sur Electron est que l'on peut utiliser plusieurs lignes :

```
DEF FN CARRE(X)= X*X
mais :
DEF FNFACT (I)
IF I = Ø THEN = 1 : ENDFN
= FNFACT(I-1)*I
ENDFN
```

LE GRAPHISME DE HAUT NIVEAU : L'INSTRUCTION VDU 23

L'Electron dispose de la possibilité laissée à l'utilisateur de créer lui-même ses propres caractères, dans une matrice 8 × 8. Chaque dessin est défini par 8 nombres, de Ø à 255, dans une instruction VDU 23,I,...
I est le code du caractère (entre 224 et 255).
Exemple : VDU 23,224,13,25,64,12,Ø,29,62,13

Les dessins peuvent être ensuite utilisés dans des PRINT, par PRINT CHR$ (I).

GRAPHISME HAUTE RESOLUTION : MODE, GCOL, PLOT

Expliquer dans ses détails les possibilités graphiques de l'Electron demanderait un livre entier. Il s'agit uniquement ici d'une approche.
Electron peut afficher en différentes résolutions que l'on détermine par l'instruction **MODE** suivant le tableau ci-dessous.

N° MODE	Ø	1	2	3	4	5	6
Résolution graphique	640 × 256	320 × 256	160 × 256	pas de graphique	320 × 256	160 × 256	pas de graphique
Résolution texte	80 × 32	40 × 32	20 × 32	80 × 25	40 × 32	20 × 32	40 × 25
Nb de couleurs	2	4	16	2	2	4	2

La plupart des programmes de ce livre (sauf mention contraire) sont réputés être en **MODE 1**.

Il est à noter que les coordonnées graphiques sont toujours indiquées dans une matrice 1 280 × 1 024. Ainsi en **MODE Ø**, 8 points ont la même coordonnée, en mode 1 16 points, etc. L'avantage est que votre repérage est compatible quel que soit le mode utilisé.

La commande **GCOL 0, x** vous permettra de positionner vos couleurs.
x = c pour la couleur du tracé
x = 128+c pour la couleur du fond.
Ainsi :
MODE 1 : GCOL 0, 1 : GCOL 0, 130
vous permet un tracé rouge sur fond jaune.

PLOT a, x, y permet le tracé effectif.
(x, y) représente bien sûr les coordonnées du point concerné.
Le paramètre a, compris entre 0 et 95, fait la force et l'intérêt du graphique Electron. Seuls quelques exemples sont donnés ici, mais vous trouverez dans le "USER GUIDE" l'explication de ces 96 valeurs possibles.

RECONNAISSANCE D'OBJETS SUR L'ECRAN LECTURE DE LA COULEUR D'UN POINT PLACE EN LIGNE Y, COLONNE X : FONCTION POINT (X, Y)

Electron dispose d'une fonction que l'on utilise dans de très nombreux jeux permettant de reconnaître la couleur d'un point donné de l'écran :
POINT (X, Y) $0 \leqslant X \leqslant 1\,279$
$0 \leqslant Y \leqslant 1\,023$
Cette fonction permet également d'adresser un caractère texte.
Chacun prenant une matrice de 32 × 32, si l'on veut connaître la couleur du point en haut à gauche du caractère en colonne I, ligne J, faire :
POINT (I × 32, 1 023 – J × 32)
de même, pour connaître la couleur du point au milieu faire :
POINT (I × 32 + 16, 1 007 – J × 32)

GESTION SONORE
ENVELOPE, SOUND

Nous ne ferons qu'effleurer les possibilités de programmation sonores de l'Electron.
Il nous est possible de déterminer 16 envelopes différentes par l'instruction **ENVELOPE** :
Elle permet de définir la courbe représentative de l'onde sonore désirée.

SOUND, V, E, H, D émet un son.
V voie de sortie
E numéro d'envelope
H hauteur du son
D Durée du son

Smokey

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Une colonne de fumée grise apparaît sur l'écran. Le joueur, représenté par un point noir, devient visible sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :
La ligne 9Ø fait apparaître la fumée au bas de l'écran ; la ligne 1ØØ permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 13Ø fait monter la colonne.
Le joueur apparaît grâce à la ligne 15Ø, et son déplacement est effectué par les lignes 18Ø et 19Ø.
Les lignes 25Ø et 26Ø se chargent de l'affichage du résultat.

POUR JOUER :
On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un rond coloré dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avaler, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :
On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```
10 REM SMOKEY
20 *FX4,1
30 VDU 23,224,0,28,62,127,127,127,62,
28
40 VDU 23,225,0,54,127,127,62,28,8,0
50 S=0
60 X=29:Y=RND(24)-1
70 U=8:V=12
80 CLS
90 COLOUR 2:PRINT TAB(Y,X);CHR$(225)
100 SOUND 1,-1,150,1
110 Y=Y-(RND(1)<.5 AND Y <24)+(RND(1)<
.5 AND Y>0)
120 T=1
130 X=X-T
140 IF X=0 THEN 250
150 COLOUR 1:PRINT TAB(V,U);CHR$(224)
160 IF T<>0 THEN A=INKEY(50)
170 PRINT TAB(V,U);" "
180 U=U-(A=138 AND T=1 AND X<24)
190 U=U-(T=0)
200 V=V-(A=137 AND V<24)+(A=136 AND V>
0)
210  C=FNPNT(V,U)
220 IF C=0 THEN 90
230 A=0:T=0:S=S+1:SOUND 1,1,150,1
240 IF U<29 THEN 150
250 IF T=0 THEN PRINT TAB(0,0);"BRAVO"
:GOTO 270
260 PRINT TAB(0,0);"PERDU"
270 PRINT TAB(10,1);"SCORE : ";S
280 *FX4,0
290 END
300 DEF FNPNT(A,B)
310 =POINT(A*32+16,1007-B*32)
>VDU12,3
```

Course de voitures

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :
L'impression de la route est assurée par la ligne 11Ø.
La ligne 12Ø fait avancer la route, et la ligne 15Ø imprime la voiture.
La ligne 18Ø contrôle son mouvement, et la ligne 19Ø détermine si la route fait ou non des virages.
La ligne 21Ø regarde si l'on a touché le bord de la route.
La ligne 23Ø incrémente le score, et la ligne 24Ø efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).
La ligne 27Ø affiche le temps avant l'accident.

POUR JOUER :
Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage le score réalisé et du temps obtenu.
On pourra arrêter le déroulement de la partie à tout moment en pressant les touches ESCAPE.

EXTENSIONS POSSIBLES :
On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```
   10 REM COURSE DE VOITURES
   20 *FX4,1
   30 VDU23,224,85,170,85,170,85,170,85,
170
   40 VDU23,225,255,255,255,255,255,255,
255,255
   50 VDU23,226,93,127,85,20,93,127,93,8
   60 X$=CHR$(224)+STRING$(5,CHR$(225))+
CHR$(224)
   70 MODE1:COLOUR 128:CLS
   80 T=0:Z=0
   90 X=30:Y=11:A=5:B=10:S=0
  100 COLOUR 2
  110 PRINT TAB(Y,X);X$
  120 PRINT
  130 F=FNPNT(B,A)
  140 COLOUR 1
  150 PRINT TAB(B,A);CHR$(226):COLOUR 12
8
  160 K=INKEY(0)
  170 C=B
  180 B=B-(K=137)+(K=136)
  190 Y=Y+((RND(2)-1) AND Y<25)-((RND(2)
-1) AND Y>3)
  200 IF Z<24 THEN Z=Z+1:TIME=0:GOTO 240
  210 IF F=0 THEN 260
  220 SOUND 1,-1,150,1
  230 S=S+1
  240 PRINT TAB(C,A);" "
  250 GOTO 100
  260 PRINT TAB(25,1);"SCORE : ";S
  270 PRINT TAB(25,3);"TEMPS : ";INT(TIM
E/100)
  280 *FX4,0
  290 END
  300 DEF FNPNT(C,D)
  310 =POINT(C*32+4,1007-D*32)
>
```

Danger spatial

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe, pour un joueur.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes aux commandes d'un engin spatial qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

LE PROGRAMME :
La position de chaque météorite est déterminée par les lignes 6Ø et 8Ø ; elle est imprimée par les lignes 7Ø et 9Ø.
L'avance des météorites est réalisée par la ligne 1ØØ.
Les lignes 1ØØ et 11Ø vérifient si vous avez ou non heurté une météorite.
La ligne 13Ø dessine votre vaisseau spatial, et la ligne 16Ø permet de le déplacer.
Le compteur se trouve en ligne 17Ø.

POUR JOUER :
Le jeu est lancé par l'ordre RUN.
Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).
On peut arrêter la partie à tout moment en tapant ESCAPE. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :
Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```
   10 REM DANGER SPATIAL
   20 VDU23,224,255,255,255,255,255,255,
255,255
   30 *FX4,1
   40 CLS
   50 X=18:Z=0:Y=X
   60 H=RND(40)-1
   70 IF RND(1)<.7 THEN COLOUR 2:PRINT T
AB(H,29);CHR$(224);:SOUND 1,-1,100,1
   80 K=RND(40)-1
   90 IF RND(1)<.6 THEN COLOUR 1:PRINT T
AB(K,29);CHR$(224);:SOUND 1,-1,105,1
  100 PRINT TAB(0,30);CHR$(10):B=FNPNT(X
,12):C=FNPNT(X+1,12)
  110 IF B=1 OR B=2 OR C=1 OR C=2 THEN 1
90
  120 PRINT TAB(Y-1,11);"    ";TAB(Y-1,1
0);"    "
  130 COLOUR 1:PRINT TAB(X,12);"##"
  140 A=INKEY(0)
  150 Y=X
  160 X=X-(A=137 AND X<38)+(A=136 AND X>
1)
  170 Z=Z+1
  180 GOTO 60
  190 PRINT TAB(0,2);"LE SCORE FINAL EST
 DE ";Z;".   "
  200 FOR I=200TO150STEP-2:SOUND1,1,I,1:
NEXT
  210 *FX4,0
  220 END
  230 DEF FNPNT(C,D)
  240 =POINT(C*32,1023-D*32)
>VDU12,3
```

Tic tac toe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un O ou un X (ici un cœur ou un carreau) dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :
Le terrain de jeu est dessiné à l'aide de la procédure PROCFEN (32Ø-36Ø) et de la boucle I (1ØØ-14Ø).
La ligne 17Ø s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 2ØØ) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 23Ø et 24Ø permettent de déplacer le symbole sur le terrain de jeu.
La ligne 26Ø vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.
La ligne 27Ø affiche le coup, et la ligne 28Ø réalise le changement de joueur.

POUR JOUER :
Le terrain de jeu sera affiché dès la frappe de RUN.
Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ←, → et ↑, ↓).
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche RETURN, le coup sera refusé si la case n'est pas libre.
Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper ESCAPE pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :
On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```
   10 REM TIC TAC TOE
   20 *FX4,1
   30 COLOUR 128:CLS
   40 VDU23,224,0,54,127,127,62,28,8,0
   50 VDU23,225,0,8,28,62,127,62,28,8
   60 VDU23,226,255,255,255,255,255,255,
255,255
   70 PROCFEN(1,1,15,3):COLOUR 128
   80 VDU26
   90 COLOUR 0
  100 FOR I=1 TO 3
  110   FOR J=1 TO 3
  120     PROCFEN(4*I-1,4*J-1,3,0)
  130     NEXT J
  140   NEXT I
  150 VDU26:COLOUR3
  160 K=0
  170 COLOUR K+1:PRINT TAB(0,25);"JOUEUR
 ";CHR$(224+K)
  180 I=1:J=1
  190 C=FNPNT(4*I,4*J)
  200 COLOUR 3-C:PRINT TAB(4*I,4*J);"X"
  210 PRINT TAB(4*I,4*J);:IF C<>0 THEN C
OLOUR C:PRINT CHR$(223+C) ELSE COLOUR 0:
PRINTCHR$(225)
  220 A=INKEY(0)
  230 J=J-(A=138 AND J<3)+(A=139 AND J>1
)
  240 I=I-(A=137 AND I<3)+(A=136 AND I>1
)
  250 PRINT TAB(25,1);C
  260 IF C<>0 OR A<>13 THEN 190
  270 COLOUR K+1:PRINT TAB(4*I,4*J);CHR$
(224+K)
  280 K=ABS(K-1)
  290 GOTO 170
  300 DEF FNPNT(X,Y)
  310 =POINT(X*32+16,1007-Y*32)
  320 DEF PROCFEN (X,Y,T,C)
  330 VDU 28,X,Y+T-1,X+T-1,Y
  340 COLOUR 128+C
  350 CLS
  360 ENDPROC
>VDU12,3
```

Horloge 2

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : simulation.
Difficulté du jeu : aucune.

LE JEU :
Ce programme simule une horloge digitale grand format en utilisant les caractères d'origine de l'Electron, notablement agrandis. Il présente la particularité de permettre la mise à l'heure, et éventuellement l'arrêt ou le départ de l'horloge.

LE PROGRAMME :
Ce programme a pour base HORLOGE.
Les lignes 35Ø à 61Ø permettent la remise à l'heure des heures (4ØØ à 49Ø) et des minutes (5ØØ à 58Ø).

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure.
Une fois l'horloge en marche, la frappe de la touche RETURN permet de la régler.
Faire ESCAPE pour arrêter le programme.

EXTENSIONS POSSIBLES :
On peut améliorer l'affichage des secondes, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment du passage de l'heure.

```
   10 REM HORLOGE 2
   20 *FX4,1
   30 CLS
   40 INPUT TAB(0,2);"HEURE (FORME HHMM)
 : " A$
   50 IF LEN(A$)<>4 THEN 40
   60 T=TIME:T1=T
   70 COLOUR 129:CLS:COLOUR 130:COLOUR 1
   80 PRINT TAB(0,23)STRING$(240," "):PR
INT TAB(0,25);"POUR REMETTRE A L'HEURE,
 TAPER 'RETURN'"
   90 K=1
  100 A=VAL(MID$(A$,K,1))
  110 IF A<>1 AND A<>4 THEN PRINT TAB(6*
K-5,1);"    "
  120 FOR J=1 TO 5
  130    IF A=0 OR (A>3 AND A<>7) THENPRI
NT TAB(6*K-5,J);" "
  140    IF A<>5 AND A<>6 THEN PRINT TAB(
6*K-3,J);" "
  150    IF A=0 OR A=2 OR A=6 OR A=8 THEN
 PRINT TAB(6*K-5,J+4);" "
  160    IF A<>2 THEN PRINT TAB(6*K-3,J+4
);" "
  170    NEXT J
  180 IF A>1 THEN PRINT TAB(6*K-4,5);"
"
  190 IF A<>1 AND A<>4 AND A<>7 THEN PRI
NT TAB(6*K-5,9);"    "
  200 K=K+1
  210 IF K<5 THEN 100
  220 B$=A$
  230 A=1+VAL(A$)
  240 IF A=2360 THEN A=0
  250 IF (A MOD 100)=60THEN A=A+40
  260 A$=RIGHT$("000"+STR$(A),4)
  270 H=1
  280 IF TIME-T>=6000THENT=TIME:GOTO70
  290 IF INKEY(0)=13 THEN 350
  300 IF TIME-T1<50 THEN 300
  310 T1=TIME
  320 PRINT TAB(24,9);RIGHT$("0"+STR$(IN
T((TIME-T)/100)),2)
  330 PRINT TAB(11,9);:IF H=1 THEN PRINT
":" ELSE PRINT" "
  340 H=ABS(H-1):GOTO280
  350 PRINT TAB(0,24);"      VOUS POUVEZ
REMETTRE A L'HEURE":PRINT"   SUCCESSIVEM
ENT HEURES PUIS MINUTES.   "
```

```
  360 PRINT"     UTILISEZ LES FLECHES <-
 ET ->":PRINT"         PUIS 'RETURN' POUR
VALIDER"
  370 Y$=LEFT$(B$,2):Z$=RIGHT$(B$,2):Y=V
AL(Y$):Z=VAL(Z$)
  380 PRINT TAB(15,17);"        ":PRINT
TAB(15,18);"        ":PRINT TAB(15,19);"
        "
  390 COLOUR128:COLOUR1:PRINT TAB(16,18)
;Y$;TAB(20,18);Z$
  400 COLOUR 3:PRINT TAB(16,18);Y$
  410 X=GET
  420 IF X<>136 THEN 450
  430 Y=Y-1:IF Y=-1 THEN Y=23
  440 Y$=RIGHT$("0"+STR$(Y),2):GOTO 400
  450 IF X<>137 THEN480
  460 Y=Y+1:IF Y=24 THEN Y=0
  470 Y$=RIGHT$("0"+STR$(Y),2):GOTO 400
  480 IF X<>13 THEN 410
  490 COLOUR 1:PRINT TAB(16,18);Y$
  500 COLOUR 3:PRINT TAB(20,18);Z$
  510 X=GET
  520 IF X<>136 THEN 550
  530 Z=Z-1:IF Z=-1 THEN Z=59
  540 Z$=RIGHT$("0"+STR$(Z),2):GOTO 500
  550 IF X<>137 THEN580
  560 Z=Z+1:IF Z=60 THEN Z=0
  570 Z$=RIGHT$("0"+STR$(Z),2):GOTO 500
  580 IF X<>13 THEN 510
  590 COLOUR 1:PRINT TAB(20,18);Z$
  600 A$=Y$+Z$
  610 GOTO 60
>VDU12,3
```

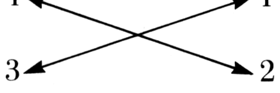
Dames

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.
Par contre le changement de joueur est réalisé à chaque coup. Sauf s'il y a prise.

LE PROGRAMME :
Les boucles I (9Ø-17Ø) et J (7Ø-18Ø) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (21Ø-27Ø) et I (22Ø-26Ø).
La boucle X (28Ø-33Ø) affiche les pions déjà pris.
La ligne 34Ø indique le joueur qui a le trait.
Le coup est entré aux lignes 37Ø à 43Ø.
La ligne 41Ø regarde si le déplacement est possible.
La ligne 46Ø traite le cas d'un déplacement et la ligne 47Ø celui d'un saut.
La ligne 53Ø assure le changement de joueur.

POUR JOUER :
Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et une forme de pion de couleur apparaît pour indiquer que le trait appartient au joueur correspondant.
On joue en indiquant par un mot de 3 lettres et la ligne la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple D91 signifie que le O situé en ligne 9, colonne D, se déplace dans la direction 1.
De même GØ3 signifiera pièce noire, ligne Ø, colonne G, direction 3.
Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant à Electron de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```
   10 REM DAMES
   20 DIM R(10,10),C(10,10),P(2)
   30 VDU23,224,0,0,0,0,0,0,0,0
   40 VDU23,225,0,28,62,127,127,127,62,2
8
   50 MODE1:COLOUR 128:COLOUR 3:CLS
   60 PRINT TAB(0,0);"   A  B  C  D  E
F  G  H  I  J"
   70 FOR J=0 TO 9
   80   PRINT TAB(0,J*3+3);J
   90   FOR I=0 TO 9
  100     K=((I+J) MOD 2)+1
  110     PROCFEN(I*3+2,J*3+2,3,K)
  120     VDU26:COLOUR 128
  130     R(I,J)=0
  140     IF K=2 THEN 170
  150     IF J<4 THEN R(I,J)=1:C(I,J)=3
  160     IF J>5 THEN R(I,J)=1:C(I,J)=0
  170     NEXT I
  180   NEXT J
  190 H=1
  200 COLOUR 129
  210 FOR J=0 TO 9
  220   FOR I=0 TO 9
  230     K=((I+J) MOD 2)+1:IF K=2 THEN
260
  240     X=3*I+3:Y=3*J+3
  250     COLOUR C(I,J):PRINT TAB(X,Y);C
HR$(224+R(I,J))
  260     NEXT I
  270   NEXT J
  280 FOR X=0TO1
  290   IF P(X)=0 THEN 330
  300   COLOUR 129:COLOUR -3*(X=0)
  310   FOR I=1 TO P(X)
  320     J=I-1:PRINT TAB(35+(J MOD 5),2
0+4*X+(J DIV 5));CHR$(225):NEXT I
  330   NEXT X
  340 COLOUR H*3:PRINT TAB(36,4);CHR$(22
5)
  350 COLOUR128:COLOUR3:PRINT TAB(33,6);
"A VOUS"
```

```
  360 PRINT TAB(35,9);"CLD":PRINT TAB(33
,12);"C:A A H";TAB(33,13);"L:0 A 9";TAB(
33,14);"D:DIREC";TAB(35,15);"4 1";TAB(36
,16);"X";TAB(35,17);"3 2"
  370 PRINT TAB(35,9);:C=GET-65:IF C<0 O
R C>9 THEN SOUND 1,-1,100,2:GOTO 370
  380 PRINTCHR$(C+65)
  390 PRINT TAB(36,9);:L=GET-48:IF L<0 O
R L>9 THEN SOUND 1,-1,100,2:GOTO 390
  400 PRINTCHR$(L+48)
  410 IF R(C,L)=0 OR (C(C,L)=0 AND H=1)
OR (C(C,L)=3 AND H=0) THEN SOUND 1,-1,10
0,2:GOTO 370
  420 PRINT TAB(37,9);:D=GET-48:IF D<1 O
R D>4 THEN SOUND 1,-1,100,2:GOTO 420
  430 PRINTCHR$(D+48)
  440 T=(1 AND (D=1 OR D=2))-(1 AND (D=4
 OR D=3))
  450 S=(1 AND (D=3 OR D=2))-(1 AND (D=4
 OR D=1))
  460 IF R(C+T,L+S)=0 THEN R(C,L)=0:R(C+
T,L+S)=1:C(C+T,L+S)=-3*(H=1):GOTO 530
  470 IF C(C+T,L+S)=-3*(H=0) AND R(C+T,L
+S)=1 AND R(C+2*T,L+2*S)=0 THEN 490
  480 SOUND 1,-1,100,2:GOTO 370
  490 R(C,L)=0:R(C+T,L+S)=0:R(C+2*T,L+2*
S)=1:C(C+2*T,L+2*S)=-3*(H=1)
  500 SOUND1,2,100,8
  510 P(H)=P(H)+1
  520 GOTO 200
  530 H=ABS(H-1)
  540 GOTO 200
  550 END
  560 DEF PROCFEN(X,Y,T,C)
  570 VDU 28,X,Y+T-1,X+T-1,Y
  580 COLOUR 128+C
  590 CLS
  600 ENDPROC
>
```

Œufs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des œufs que vous devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :
L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 15Ø et 19Ø ; son déplacement aléatoire est géré par le sous-programme 28Ø-42Ø (en particulier les lignes 38Ø à 41Ø).
La ligne 2ØØ permet de reboucler le programme sur la ligne 15Ø.
Le déplacement du joueur est régi par les lignes 22Ø et 23Ø, et le test permettant de détecter la présence d'un œuf est effectué en ligne 24Ø.
La ligne 26Ø permet l'affichage du score.
La fin de jeu est détectée en ligne 31Ø.

POUR JOUER :
On lancera le programme par RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr améliorer la couleur, les sons, le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'œufs.

```
   10 REM OEUFS
   20 *FX4,1
   30 VDU 23,224,0,28,62,127,127,127,62,
28
   40 VDU23,225,0,62,34,127,127,127,255,
28
   50 VDU23,226,0,0,0,64,120,14,3,0
   60 VDU23,227,0,0,0,1,14,48,224,0
   70 VDU23,228,0,0,60,34,98,2,3,0
   80 VDU23,229,0,0,30,34,35,32,224,0
   90 VDU23,230,0,127,127,127,127,127,12
7,127
  100 X$=CHR$(226)+CHR$(225)+CHR$(227)
  110 Y$=CHR$(228)+CHR$(225)+CHR$(229)
  120 CLS:P=200:R=18:S=10:T=0
  130 COLOUR 3:PRINT TAB(0,25);"TEMPS :
";P
  140 X=RND(36)-1:Y=RND(21)-1
  150 COLOUR 1:PRINT TAB(X,Y);X$
  160 A=INKEY(0):IF A<>-1 THEN GOSUB 210
  170 *FX15,1
  180 COLOUR 2:GOSUB 280
  190 COLOUR 1:PRINT TAB(X,Y);Y$
  200 GOSUB 280:GOTO 150
  210 PRINT TAB(R,S);" "
  220 R=R-(A=137 AND R<37)+(A=136 AND R>
0)
  230 S=S-(A=138 AND S<21)+(A=139 AND S>
0)
  240 G=FNPNT(R,S)
  250  IF G=2 THEN T=T+1:SOUND 1,1,100,2
  260 COLOUR 3:PRINT TAB(15,25);"SCORE :
 ";T
  270 RETURN
  280 A=X:B=Y
  290 P=P-1
  300 PRINT TAB(8,25);P;" "
  310 IF P>0 THEN 350
  320 *FX4,0
  330 SOUND 1,2,100,12
  340 END
  350 COLOUR 3:PRINT TAB(R,S);CHR$(230)
  360 SOUND1,-1,100,1
  370 COLOUR 2:PRINT TAB(A,B);" "+CHR$(2
24)+" "
  380 X=ABS(X+INT(1+2*(RND(1)<.5)))
  390 Y=ABS(Y+INT(1+2*(RND(1)<.5)))
  400 IF X>36 THEN X=1
  410 IF Y>21 THEN Y=1
  420 RETURN
  430 DEF FNPNT(X,Y)
  440 =POINT(X*32+16,1007-Y*32)
>VDU12,3
```

Taquin de lettres

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.
Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case blanche en flash vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :
La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 7Ø.
L'impression en est réalisée par la boucle I (9Ø-13Ø).
Le mélange des lettres est fait par la boucle I (14Ø-22Ø).
La ligne 26Ø réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 31Ø et 32Ø en utilisant les touches I, J, K, M.
La ligne 36Ø réimprime la lettre concernée à l'emplacement du carré coloré.
La ligne 39Ø comptabilise le nombre d'essais qui est imprimé grâce à la ligne 25Ø.

POUR JOUER :
On lance le programme par RUN.
La partie est facilitée par l'affichage du proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de curseur.
Le nombre d'essais est affiché en permanence sous le puzzle.

EXTENSIONS POSSIBLES :
Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```
10 REM JEU94
20 REM TAQUIN DE LETTRES
30 *FX4,1
40 N=3
50 COLOUR 128:CLS
60 PROCFEN(0,0,N*2+1,2)
70 PROCFEN(1,1,N*2-1,1)
80 PROCFEN(20,10,N+2,3)
90 A$=""
100 R$="ABCDEFGH"
110 S$=R$+" "
120 FOR I=0 TO N*N-2
130   F$=MID$(R$,I+1,1)
140   X=I MOD N:Y=I DIV N
150   COLOUR 128:COLOUR 3:PRINT TAB(X+
21,Y+11);F$
160   NEXT I
170 FOR I=1 TO N*N-1
180   X=RND(N*N-1)
190   IF MID$(R$,X,1)="-" THEN 180
200   A$=A$+MID$(R$,X,1)
210   R$=LEFT$(R$,X-1)+"-"+MID$(R$,X+1
)
220   F$=MID$(A$,I,1)
230   X=I MOD N:Y=I DIV N
240   COLOUR 129:COLOUR 2:PRINT TAB(X*
2+1,Y*2+1);F$
250   NEXT I
260 A$=" "+A$
270 X=0:Y=0:S=0:C=32
280 COLOUR 128:COLOUR 3:PRINT TAB(0,25
);"NOMBRE D'ESSAIS : ";S
290 COLOUR 128:PRINT TAB(X*2+1,Y*2+1);
" "
300 PRINT TAB(0,28);A$
310 T=INKEY(0)
320 IF T=-1 THEN COLOUR 129:PRINT TAB(
X*2+1,Y*2+1);" ":GOTO 290
330 I=X:J=Y
340 X=X-(T=137 AND X<N-1)+(T=136 AND X
>0)
350 Y=Y-(T=138 AND Y<N-1)+(T=139 AND Y
>0)
360 IF X=I AND Y=J THEN SOUND 1,1,200,
2: GOTO 280
370 L=Y*N+X+1
380 M=J*N+I+1
390 COLOUR 129:COLOUR 2:PRINT TAB(I*2+
1,J*2+1);MID$(A$,L,1)
```

```
  400 A$=LEFT$(A$,M-1)+MID$(A$,L,1)+MID$
(A$,M+1)
  410 A$=LEFT$(A$,L-1)+" "+MID$(A$,L+1)
  420 S=S+1
  430 IF A$<>S$ THEN 280
  440 PRINT TAB(0,28);STRING$(40," "):PR
INT TAB(0,28);"BRAVO: GAGNE EN ";S;" ESS
AIS."
  450 *FX4,0
  460 END
  470 DEF PROCFEN(X,Y,T,C)
  480 VDU28,X,Y+T-1,X+T-1,Y
  490 COLOUR 128+C:CLS
  500 VDU26:COLOUR 128
  510 ENDPROC
>VDU12,3
```

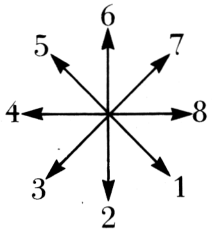
Quatre à quatre

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (un ou deux joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :
La variable alphanumérique A$, en ligne 5Ø, contient la table des coups possibles (déplacement du cavalier).
La boucle I (7Ø-11Ø) dessine le terrain de jeu, et les positions de départ.
La ligne 13Ø permet d'afficher O ou X, selon le joueur qui a le trait.
La ligne 16Ø permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.
Les lignes 19Ø et 2ØØ permettent le déplacement du curseur.
La ligne 21Ø regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 22Ø réimprime la figure qui était sous le curseur. La ligne 25Ø réalise le contrôle de départ.
Les lignes 27Ø et 28Ø effectuent le déplacement. La ligne 31Ø réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 32Ø, le changement de joueur est réalisé par la ligne 34Ø et un "." est imprimé à l'emplacement de départ. La fonction 37Ø-38Ø est utilisée pour les divers contrôles.

POUR JOUER :
On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.
Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.
Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant à l'Electron de participer au jeu avec de bonnes chances de l'emporter.

```
   10 REM QUATRE A QUATRE
   20 *FX4,1
   30 CLS:VDU 23,224,0,28,62,127,127,127
,62,28:VDU 23,225,0,129,66,36,24,36,66,1
29
   40 PROCFEN (0,0,18,3):VDU 26
   50 H=1:A$="5665322356"
   60 COLOUR 128:COLOUR 3
   70 FOR I=1 TO 5
   80   COLOUR 3:PRINT TAB(I*3+1,1);I:PR
INT TAB(1,I*3+1);I
   90   FOR J=1 TO 5
  100     PRINT TAB(I*3+1,J*3+1);:IF (I>
=3 AND J<=3) OR (I<=3 AND J>=3)THEN COLO
UR 3:PRINT "." ELSE COLOUR 1-(I<3):PRINT
 CHR$(224-(I>3))
  110     NEXT J,I
  120   X=1:Y=1
  130   COLOUR 2-H:PRINT TAB(0,25);"JOUE
UR ";CHR$(224+H)
  140   J=FNPNT(X*3+1,Y*3+1)
  150   I=J
  160   PRINT TAB(X*3+1,Y*3+1);" "
  170   A=X:B=Y
  180   R=INKEY(0)
  190   Y=Y-(R=138 AND Y<5)+(R=139 AND Y
>1)
  200   X=X-(R=137 AND X<5)+(R=136 AND X
>1)
  210   IF R>48 AND R<57 THEN 250
  220   IF J=0 THEN COLOUR 3:PRINT TAB(3
*A+1,3*B+1);".":GOTO 130
  230   COLOUR J:PRINT TAB(3*A+1,3*B+1);
CHR$(226-J)
```

```
  240    GOTO 130
  250     IF I+H<>2 THEN 230
  260    J=R-48
  270    S=Y:Y=Y+VAL(MID$(A$,J,1))-4
  280    T=X:X=X+VAL(MID$(A$,J+2,1))-4
  290    IF X<1 OR X>5 OR Y<1 OR Y>5 THEN
 Y=S:X=T:A=T:B=S:J=I:GOTO 230
  300    K=FNPNT(X*3+1,Y*3+1)
  310    IF K<>0 THEN Y=S:X=T:A=T:B=S:J=I
:GOTO 230
  320    COLOUR3:PRINT TAB(3*T+1,3*S+1);"
."
  330    COLOUR I:PRINT TAB(3*X+1,3*Y+1);
CHR$(226-I)
  340    H=ABS(H-1):SOUND 1,2,100,8
  350    GOTO130
  360    END
  370    DEF FNPNT(X,Y)
  380    =POINT(X*32+16,1007-Y*32)
  390    DEF PROCFEN(X,Y,T,C)
  400    VDU 28,X,Y+T-1,X+T-1,Y
  410    COLOUR 128+C
  420    CLS
  430    ENDPROC
>VDU12,3
```

Cibles

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard.
Difficulté du jeu : facile.

LE JEU :
Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 9) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :
Le nombre de cibles est entré en ligne 5Ø. La boucle I (9Ø-11Ø) effectue l'impression des cibles.
La ligne 13Ø affichera le score en permanence.
La balle est affichée par la ligne 14Ø.
La boucle I (16Ø-22Ø) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 21Ø).
La ligne 24Ø regarde si l'on a tiré, et la ligne 25Ø regarde si la boule est encore sur la ligne de départ.
La ligne 26Ø fait progresser la boule d'une case vers la droite, et la ligne 26Ø vérifie si l'on a ou non atteint les cibles ; puis la ligne 28Ø regarde ce que l'on a touché.
La ligne 31Ø incrémente le score.
Les lignes 32Ø et 33Ø s'occupent de la fin de partie.

POUR JOUER :
Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de RETURN. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant RETURN.
Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant ESCAPE, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :
On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```
   10 REM CIBLES
   20 DIMV(9)
   30 COLOUR 128:COLOUR 3:CLS
   40 VDU23,224,0,28,62,127,127,127,62,2
8
   50 INPUT TAB(9,1);"NOMBRE DE CIBLES :
 " N
   60 P=N+1:J=30:S=0:B=1
   70 *FX4,1
   80 CLS:COLOUR 129
   90 FOR I=1 TO N
  100    PRINT TAB(J,3*I-1);"   ":PRINT T
AB(J+2,3*I);" ":PRINT TAB(J,3*I+1);"   "
  110    NEXT I
  120 K=0:A=18
  130 COLOUR 128:COLOUR 3:PRINT TAB(0,28
);"SCORE : ";S;"   BALLE : ";B
  140 COLOUR 2:PRINT TAB(K,A);CHR$(224)
  150 D=A
  160 FOR I=1 TO N
  170    U=RND(P)
  180    V(I)=U
  190    COLOUR131:COLOUR0:PRINT TAB(J+1,
3*I);CHR$(U+48)
  200    X=INKEY(0)
  210    A=A-(X=138)+(X=139)
  220    NEXT I
  230 COLOUR 128:PRINT TAB(K,D);" "
  240 IF X=13 AND K=0 THEN B=B+1:GOTO 26
0
  250 IF K=0 THEN 140
  260 K=K+1
  270  IF K<J THEN 140
  280 IF (K=J OR K=J+1) AND (A MOD 3)<>0
 THEN COLOUR 3:PRINT TAB(0,30);"===   PE
RDU":GOTO 340
  290 IF K=J THEN 140
  300 X=0:IF K=J+1 THEN X=V(A DIV 3)
  310 S=S+X
  320 P=P-1
  330 IF P>1 THEN 120
  340 COLOUR 3
  350 PRINT TAB(8,28);S
  360 *FX4,0
>VDU12,3
```

Gouttes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un rond gris, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :
Les deux boucles emboîtées I (5Ø-13Ø) et J (6Ø-12Ø) dessinent le récipient et les obstacles.
La ligne 17Ø dessine la goutte et la ligne 21Ø envoie dans un sous-programme (37Ø-4ØØ) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 29Ø, et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 23Ø) selon la valeur définie pour K en ligne 15Ø.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 24Ø, 25Ø ou 26Ø).
Selon le résultat du test, on utilisera 27Ø, 29Ø, 31Ø ou 34Ø.

POUR JOUER :
Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

```
   10 REM GOUTTES
   20 MODE 2
   30 VDU23,224,255,255,255,255,255,255,
255,255
   40 VDU 23,225,0,28,62,127,127,127,62,
28
   50 FOR I=12 TO 20
   60   FOR J=2 TO 17
   70     COLOUR 7
   80     PRINT TAB(1,I);CHR$(224)
   90     PRINT TAB(18,I);CHR$(224)
  100     PRINT TAB(J,20);CHR$(224)
  110     IF I<>12 AND I<>20 AND RND(1)>
.75THEN COLOUR 1:PRINT TAB(J,I);CHR$(224
)
  120     NEXT J
  130   NEXT I
  140 Y=RND(8)+5:X=6
  150 K=-2*(RND(1)<.5)-1
  160 L=RND(5)+1:I=0
  170 COLOUR L:PRINT TAB(Y,X);CHR$(225)
  180 FOR I=1 TO 20:NEXT
  190 PRINT TAB(Y,X);" "
  200 U=X+1:V=Y
  210 GOSUB 370
  220 IF J=0 THEN 290
  230 U=X:V=Y+K
  240 GOSUB 370
  250 IF J=0 THEN 310
  260 IF I=0 THEN 340
  270 COLOUR L:PRINT TAB(Y,X);CHR$(225)
  280 GOTO 140
  290 X=X+1
  300 GOTO 150
  310 Y=Y+K
  320 I=1
  330 GOTO 170
  340 K=-K
  350 I=1
  360 GOTO 230
  370 C=POINT(64*V+32,1007-32*U)
  380 SOUND 1,-1,115-U,1
  390 J=-(C<>0):IF U=24 THEN PRINT TAB(7
,29);"C'EST FINI...":END
  400 RETURN
>VDU12,3
```

Obstacles

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : jeu de réflexe (pour un joueur).
Difficulté du jeu : moyenne.

LE JEU :
Des piétons se déplacent sur l'écran pour venir à la rencontre d'une petite voiture qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.
La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un piéton.

LE PROGRAMME :
La ligne 8Ø fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 7Ø.
Les lignes 8Ø, 9Ø, 1ØØ fabriquent la route (les lignes 9Ø et 1ØØ bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).
La ligne 12Ø affiche le score.
La ligne 14Ø permettra de déplacer le véhicule.
Les lignes 15Ø à 17Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté en ligne 16Ø.

POUR JOUER :
On lance le programme en tapant RUN.
La route et les piétons apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).
La rencontre avec un piéton ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant les touches ESCAPE.

EXTENSIONS POSSIBLES :
Il sera assez facile d'améliorer le dessin de la voiture, de la route, des piétons. On pourra améliorer le son et la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```
   10 REM OBSTACLES
   20 VDU23,224,93,127,85,20,93,127,93,8
   30 VDU23,225,8,20,8,62,42,8,28,54
   40 VDU23,226,255,255,255,255,255,255,
255,255
   50 CLS:S=0:Y=5:Z=5:*FX4,1
   60 COLOUR 3:FOR I=0 TO 29:PRINT TAB(0
,I);CHR$(226);"        ";CHR$(226):NEXT
I
   70 COLOUR 1:PRINT TAB(Z,3);" ";TAB(Y,
4);CHR$(224)
   80 COLOUR 3:PRINT TAB(0,29);CHR$(226)
;"        ";CHR$(226);
   90 IF RND(1)<.7 THEN COLOUR 2:PRINT T
AB(RND(8),29);CHR$(225);
  100 IF RND(1)<.8 THEN COLOUR 2:PRINT T
AB(RND(8),29);CHR$(225);
  110 PRINT TAB(39,31):SOUND1,-1,100,1
  120 PRINT TAB(15,0);"SCORE : ";S
  130 A=INKEY(0)
  140 Z=Y:Y=Y-(A=137 AND Y<9)+(A=136 AND
  Y>0)
  150 C=POINT(32*Y+16,879)
  160 S=S+1
  170 IF C=0 THEN 70
  180 *FX4,0
>VDU12,3
```

Captures dans l'espace

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

LE JEU :
Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.
Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :
La ligne 8Ø permet de faire apparaître les météorites.
La ligne 1ØØ fera apparaître le minerai d'urane et assure déplacement des météorites sur l'écran.
Les lignes11Ø à 13Ø contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane ?).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par les lignes 14Ø et 15Ø et son déplacement est contrôlé par la ligne 17Ø.

POUR JOUER :
On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.
Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.
On pourra aussi arrêter à tout moment en tapant ESCAPE.

EXTENSIONS POSSIBLES :
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.
On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
   10 REM CAPTURES DANS L'ESPACE
   20 CLS:*FX4,1
   30 VDU23,224,0,73,28,54,119,54,28,73
   40 VDU23,225,0,28,28,127,127,127,28,2
8
   50 VDU23,226,0,127,65,85,73,85,65,127
   60 X=15:Y=X:Z=0
   70 S=RND(40)-1
   80 COLOUR 1:PRINT TAB(S,31);CHR$(225)
;:SOUND 1,-1,100,1
   90 S=RND(39)-1
  100 IF RND(1)<.7 THEN COLOUR 2:PRINT T
AB(S,31);CHR$(224):SOUND 1,-1,200,1
  110  C=POINT(X*32+16,633)
  120 IF C=1 THEN 190
  130 IF C=2 THEN SOUND 1,1,100,5:Z=Z+1
  140 PRINT TAB(Y-1,11);"    "
  150 COLOUR 3:PRINT TAB(X-1,12);" ";CHR
$(226);" "
  160 A=INKEY(0):IF A=-1 THEN 70
  170 Y=X:X=X-(A=137 AND X<39)+(A=136 AN
D X>1)
  180 GOTO 70
  190 PRINT TAB(0,31):PRINT"SCORE OBTENU
 : ";Z
  200 *FX4,0
>VDU12,3
```

Anomalie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :
La boucle I (5Ø-8Ø) imprime les chiffres de repérage des grilles.
La boucle principale I (1ØØ-16Ø) imprime les symboles des deux grilles, dans un ordre différent.
La ligne 11Ø choisit le caractère à imprimer.
Le sous-programme (24Ø-3ØØ) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
Les lignes 17Ø et 18Ø changent, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées aux lignes 19Ø et 2ØØ, et le contrôle est effectué par les lignes 21Ø et 22Ø qui réalisent aussi l'affichage du résultat.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).
On entre alors son numéro de ligne, suivi de RETURN, et son numéro de colonne, suivi de RETURN.
Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.

```
   10 REM ANOMALIE
   20 COLOUR 128:MODE1:CLS:T=5
   30 PROCFEN(0,0,13,1):PROCFEN(17,0,13,
2)
   40 PROCFEN(3,3,9,2):PROCFEN(20,3,9,1)
   50 FOR I=1 TO 5
   60   COLOUR 2:PRINT TAB(I*2+1,1);I:PR
INT TAB(1,I*2+1);I
   70   COLOUR 1:PRINT TAB(18+I*2,1);I:P
RINT TAB(18,I*2+1);I
   80   NEXT I
   90 VDU19,1,10,0,0,0,19,2,12,0,0,0
  100 FOR I=1 TO 22
  110   D=RND(26)
  120   C=1:S=T
  130   GOSUB 240
  140   C=2
  150   GOSUB 240
  160   NEXT I
  170 S=0
  180 GOSUB 240
  190 VDU20:INPUT TAB(0,16);"LIGNE : " I
  200 INPUT TAB(15,16);"COLONNE : " J
  210  IF J*2+18=V AND 2*I+1=U THEN PRIN
T TAB(0,20)"====  BRAVO..." ELSE PRINT T
AB(0,20);"====  PERDU..."
  220 PRINT TAB(0,22);"LA LETTRE A TROUV
ER ETAIT ";CHR$(D+64):PRINT" LIGNE ";(U-
1)/2;" COLONNE ";(V-18)/2
  230 END
  240 CF=3-C
  250 COLOUR C:COLOUR 128
  260 U=2*RND(T)+1:V=2*INT(RND(1)*S)+17*
C-14
  270 K=POINT(32*V+16,1007-32*U)
  280 IF K<>CF AND S<>0 THEN 260
  290 PRINT TAB(V,U);CHR$(D+64)
  300 RETURN
  310 DEF PROCFEN(X,Y,T,C)
  320 VDU28,X,Y+T-1,X+T-1,Y
  330 COLOUR 128+C:CLS
  340 COLOUR 128:VDU26
  350 ENDPROC
>
```

Plusieurs en un

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : variable, selon l'utilisation.

LE JEU :
Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :
Les caractères graphiques correspondant aux symboles O et X sont définis aux lignes 5Ø et 6Ø.
Le terrain de jeu est délimité par un cadre mis en place aux lignes 8Ø et 9Ø, suivant la ligne 7Ø.
Le changement de joueur est effectué par la ligne 27Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (ligne 12Ø).
Les lignes 2ØØ et 21Ø permettent aux deux joueurs de déplacer le symbole sur le terrain.
La ligne 22Ø contrôle si le déplacement est possible (dépassement des limites).
Les lignes 25Ø et 26Ø permettent d'écrire sur une case libre.

POUR JOUER :
RUN provoque l'affichage des limites du terrain et du symbole du joueur, ainsi que des caractères graphiques correspondants que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche RETURN pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît et celui-ci procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper ESCAPE.

EXTENSIONS POSSIBLES :
On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO.

```
  10REM PLUSIEURS EN UN
  20*FX4,1
  30COLOUR 128:CLS
  40VDU23,224,255,255,255,255,255,255,2
55,255
  50VDU23,225,0,73,28,54,119,54,28,73
  60VDU23,226,0,127,65,85,73,85,65,127
  70L1=2:C1=2:L2=25:C2=37
  80VDU28,C1,L2,C2,L1:COLOUR 131:CLS
  90VDU28,C1+1,L2-1,C2-1,L1+1:COLOUR 12
8:CLS
 100VDU26
 110C=1:I=INT((C1+C2)/2):J=INT((L1+L2)/
2)
 120COLOUR 128:COLOUR C:PRINT TAB(0,0);
"JOUEUR ";CHR$(224+C)
 130D=POINT(32*I+4,1019-32*J)
 140COLOUR D:COLOUR 128:PRINT TAB(I,J);
CHR$(224+D)
 150K=INKEY(0):IF K<>-1 THEN 180
 160COLOUR 3-D:COLOUR 128+D:PRINT TAB(I
,J);CHR$(224+D)
 170GOTO140
 180IF K=13 THEN 250
 190X=I:Y=J
 200I=I-(K=137)+(K=136)
 210J=J-(K=138)+(K=139)
 220IF I=C1 OR I=C2 OR J=L1 OR J=L2 THE
N I=X:J=Y:SOUND 1,-1,100,2:GOTO 160
 230COLOUR D:COLOUR 128:PRINT TAB(X,Y);
CHR$(224+D)
 240GOTO 130
 250IF D<>0 THEN SOUND 1,-1,100,2:GOTO
160
 260COLOUR C:COLOUR 128:PRINT TAB(I,J);
CHR$(224+C)
 270C=3-C
 280GOTO 120
 290END
 300DEF PROCFEN(X,Y,T,C)
 310VDU28,X,Y+T-1,X+T-1,Y
 320COLOUR 128+C:CLS
 330COLOUR 128:VDU26
 340ENDPROC
>VDU12,3
```

Parcours du cavalier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :
La ligne 40 définit les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; la boucle emboîtée I (60-90) dessine le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 100 à 130.
La ligne 140 imprime le numéro du coup.
La ligne 150 permet le tirage au hasard d'un déplacement du cavalier.
La boucle Z (160-220) cherche si le déplacement est possible : si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 180 et 190 effectuent le déplacement.
La ligne 200 contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 210 envoie en ligne 290 si le coup est possible.
La ligne 230 gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 240 à 280 gèrent une nouvelle partie.
Les lignes 290 à 310 réalisent le coup et incrémentent le score.

POUR JOUER :
On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche : on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), puis son numéro de colonne. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par ESCAPE.

EXTENSIONS POSSIBLES :
On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers, on peut aussi améliorer considérablement le graphisme. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

```
   10 REM PARCOURS DU CAVALIER
   20 COLOUR 128:CLS
   30 PROCFEN(1,1,29,1):PROCFEN(4,4,25,2
)
   40 A$="56653223":B$="653223"+A$+"56":
A$=A$+A$
   50 S=1
   60 FOR I=1 TO 8
   70   COLOUR 130:COLOUR 0:PRINT TAB(2,
I*3+3);I:PRINT TAB(I*3+3,2);I
   80   COLOUR 128:FOR J= 1 TO 8:PRINT T
AB(J*3+2,I*3+3);"   "
   90     NEXTJ,I
  100   COLOUR 3:PRINT TAB(32,8);"LIGNE
:";TAB(34,10);".":I=GET-48:IF I<1 OR I>8
 THEN 100
  110   PRINT TAB(34,10);I
  120   PRINT TAB(31,13);"COLONNE :";TAB
(34,15);".":J=GET-48:IF J<1 OR J>8 THEN
120
  130   PRINT TAB(34,15);J
  140   COLOUR 129:COLOUR 0:PRINT TAB(I*
3+2,J*3+3);RIGHT$(" "+STR$(S),2)
  150   Z=RND(8)
  160   FOR Z=Z TO Z+8
  170     SOUND1,-1,200,1
  180     A=I+VAL(MID$(A$,Z,1))-4
  190     B=J+VAL(MID$(B$,Z,1))-4
  200     IF A<1 OR B<1 OR A>8 OR B>8 TH
EN 220
  210     IF POINT(32*(3*A+2),1023-32*(3
*B+3))=0 THEN Z=Z+8:NEXTZ:GOTO 290
  220   NEXT Z
  230 PRINT TAB(32,18);"TOTAL :";TAB(34,
20);S
  240 COLOUR 3:PRINT TAB(32,23);"AUTRE "
;
  250 K$=GET$:IF K$="O" THEN COLOUR 128:
FOR I=8 TO 23:PRINT TAB(31,I);"
":NEXT I:GOTO 50
  260 IF K$<>"N" THEN 250
  270 PRINT TAB(0,30)
  280 END
  290 I=A:J=B
  300 SOUND1,1,200,2
  310 S=S+1
  320 GOTO 140
  330 END
  340 DEF PROCFEN(X,Y,T,C)
  350 VDU28,X,Y+T-1,X+T-1,Y
  360 COLOUR 128+C:CLS
  370 COLOUR 128:VDU26
  380 ENDPROC
```

Mur de briques

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).
Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.
La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.
Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :
La ligne 2Ø définit la forme des briques, de la raquette et du mur.
La boucle I (5Ø-8Ø) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (9Ø-13Ø) et J (1ØØ-12Ø).
La ligne 15Ø dessine la raquette dont la position est contrôlée par la ligne 17Ø.
Les lignes 22Ø à 25Ø regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 26Ø à 29Ø et 3ØØ à 33Ø.
La ligne 19Ø affiche le score en permanence.

POUR JOUER :
Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).
Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.
On peut aussi terminer à tout moment par ESCAPE.

EXTENSIONS POSSIBLES :
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```
   10 REM MUR DE BRIQUES
   20 VDU23,224,0,127,127,65,65,65,127,1
27:VDU23,225,255,255,255,255,255,255,255
,255:VDU23,226,0,127,127,127,0,0,0,0
   30 X$=" "+CHR$(226)+CHR$(226)+CHR$(22
6)+" "
   40 *FX4,1:COLOUR 128:COLOUR 3:CLS
   50 FOR I=1 TO 18
   60   PRINT TAB(9+I,0);CHR$(225)
   70   PRINT TAB(10,I);CHR$(225);TAB(27
,I);CHR$(225)
   80   NEXT I
   90 COLOUR1:FOR I=1 TO 7
  100   FOR J=11 TO 26
  110     PRINT TAB(J,I);CHR$(224)
  120     NEXT J
  130   NEXT I
  140 H=-1:K=-.5:J=18:I=17:Y=J:T=0:S=0
  150 COLOUR 2:PRINT TAB(Y-2,19);X$
  160 A=INKEY(20)
  170 Y=Y-(A=137 AND Y<28)+(A=136 AND Y>
8)
  180 IF T=1 THEN PRINT TAB(J,I);" "
  190 PRINT TAB(0,20);"SCORE : ";S
  200 IF I>22 THEN 340
  210 I=I+H:J=J+K
  220 C=POINT(32*INT(J)+4,1019-32*INT(I)
)
  230 IF C=0 THEN COLOUR 2:PRINT TAB(J,I
);"*":T=1:GOTO 150
  240 IF C=3 THEN SOUND 1,1,100,1:GOTO 2
80
  250 IF C=1 THEN F=I:COLOUR 2:PRINT TAB
(J,I);"*":SOUND 1,1,100,2:T=1:GOTO 300
  260 SOUND 1,1,200,1:H=-H
  270 K=RND(1)-RND(1)
  280 K=-K:T=0
  290 GOTO210
  300 S=S+9+F
  310 K=-(K-(2*(RND(1)<.5)+1)*RND(1)/3)
  320 H=-H
  330 GOTO 150
  340 *FX4,0
  350 END
>VDU12,3
```

Echecs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion, pour deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions et les pièces sont dessinés à l'aide des caractères graphiques (VDU 23). La couleur permet de distinguer chaque camp.

LE PROGRAMME :
La boucle I (100-170) dessine l'échiquier. Les lignes 250 à 310 définissent les conditions initiales, à l'aide de deux tableaux : R (I, J) pour les codes des pièces, et C (I, J) pour la couleur.
Les boucles emboîtées J (330-380) et I (340-370) dessinent les pièces.
Les lignes 430 à 470 permettent l'entrée de la ligne et de la colonne de départ.
Les lignes 480 à 520 permettent d'entrer colonne et ligne d'arrivée.
Les lignes 530 à 560 gèrent le coup du joueur.
La ligne 570 assure le changement de joueur.

POUR JOUER :
On lance l'exécution par RUN.
L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR BLANC suivie de ENTRER VOTRE COUP ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple puis la colonne et la ligne d'arrivée. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.
Pour arrêter, il suffit de taper un slash (/).

EXTENSIONS POSSIBLES :
On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette à Electron de se comporter en partenaire honorable Bon courage !

```
   10 REM ECHECS
   20 DIMR(8,8),C(8,8)
   30 MODE 1
   40 DIMC$(2)
   50 C$(0)="BLANC":C$(1)="ROUGE"
   60 CLS
   70 PRINT
   80 PRINT"    A   B   C   D   E   F   G   H"
   90 FOR I=1TO8
  100   FORJ=1TO3
  110     IF J=2 THEN PROCNORM (3,0):PRI
NT;I
  120     FORK=1TO4
  130       PROCNORM (0,1) :PRINT TAB((K
*6)+(I MOD 2)*3-3,I*3+J);"   "
  140       PROCNORM (0,3) :PRINT TAB((K
*6)+((I+1) MOD 2)*3-3,I*3+J)"   "
  150       NEXTK
  160     NEXTJ
  170   NEXTI
  180 PROCNORM(3,0)
  190 VDU 23,224,0,16,40,68,108,56,124,0
  200 VDU 23,225,0,16,56,16,56,68,56,0
  210 VDU 23,226,0,84,124,56,56,124,124,
0
  220 VDU 23,227,0,84,40,16,108,124,124,
0
  230 VDU 23,228,0,0,16,56,56,16,124,0
  240 VDU 23,229,0,16,56,120,24,56,124,0
  250 FORI=1TO8:FORJ=1TO8
  260     IF J=2 OR J=7 THEN R(I,J)=4 EL
SE R(I,J)=6
  270     C(I,J)=0:IF J=1 OR J=2 THEN C(
I,J)=1
  280     IF J=7 OR J=8 THEN C(I,J)=3
  290     NEXT J: NEXT I
  300   R(1,1)=2:R(2,1)=5:R(3,1)=0:R(4,1
)=3:R(5,1)=1:R(6,1)=0:R(7,1)=5:R(8,1)=2
  310 FORI=1TO8:R(I,8)=R(I,1):NEXT
  320 H=1
  330  FORJ=1TO8
  340   FORI=1TO8
  350     IF C(I,J)=0 THEN370
  360     PROCNORM(C(I,J),0):PRINTTAB(I*
3+1,J*3+2);CHR$(224+R(I,J))
  370     NEXTI
  380   NEXTJ
  390 H=0
  400 COLOUR (1-2*(H=0)):PRINT TAB(28,4)
;"JOUEUR ";C$(H)
```

```
  410 PRINT TAB(29,6);"VOTRE COUP":PRINT
 TAB(29,7);"EX : E2 E4"
  420 PRINT TAB(31,10);".. .."
  430 PRINT TAB(31,10);:K$=GET$:PRINTK$:
IF K$="/" THEN PRINT TAB(0,30);:END
  440 IF K$<"A" OR K$>"H" THEN PRINT CHR
$(7);:GOTO 430
  450 PRINT TAB(32,10);:L$=GET$:PRINTL$:
IF ASC(L$)=127 THEN 430
  460 IF L$<"1" OR L$>"8" THEN PRINTCHR$
(7);:GOTO 450
  470 I=VAL(L$):J=ASC(K$)-64:IF C(J,I)<>
(1-2*(H=0)) THEN PRINT CHR$(7);:GOTO 430

  480 PRINT TAB(34,10);:K$=GET$:PRINTK$:
IF ASC(K$)=127 THEN430
  490 IF K$<"A" OR K$>"H" THEN PRINTCHR$
(7);:GOTO 480
  500 PRINT TAB(35,10);:L$=GET$:PRINTL$:
IF ASC(L$)=127 THEN 480
  510 IF L$<"1" OR L$>"8" THEN PRINTCHR$
(7);:GOTO 500
  520 I1=VAL(L$):J1=ASC(K$)-64
  530 IF C(J1,I1)=(1-2*(H=0)) THEN PRINT
 CHR$(7);:GOTO 480
  540 GOSUB 590
  550 R(J1,I1)=R(J,I):C(J1,I1)=C(J,I):R(
J,I)=6:C(J,I)=0
  560 COLOUR 1-2*(H=0):PRINT TAB(J1*3+1,
I1*3+2);CHR$(224+R(J1,I1))
  570 H=ABS(H-1)
  580 GOTO 400
  590 IF ((I+J) MOD 2)=0 THEN COLOUR 131
 ELSE COLOUR 129
  600 PRINT TAB(J*3+1,I*3+2);"  "
  610 COLOUR 128
  620 RETURN
  630  DEF PROCNORM(CL,FO)
  640 COLOUR CL : COLOUR 128+FO
  650 ENDPROC
>WIDTH0
>VDU12,3
```

Calculatrice

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : simulation d'une calculatrice.
Difficulté du jeu : moyenne.

LE JEU :
Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

LE PROGRAMME :
Les variables B$ et C$, des lignes 5Ø et 6Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format d'après le caractère graphique défini en 4Ø.
Les lignes 9Ø à 18Ø s'occupent de la gestion des entrées.
La boucle I (19Ø-22Ø) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.
Le point décimal et le signe − sont gérés par les lignes 23Ø et 24Ø.
Le nombre total des symboles entrés est contrôlé par la ligne 25Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 3ØØ à 42Ø (l'automatisme est géré par la ligne 26Ø, dans le programme principal).
Les sous-programmes placés aux lignes 43Ø-44Ø, 45Ø-46Ø, 47Ø-48Ø et 49Ø-5ØØ effectuent l'opération indiquée, selon la valeur de Z (ligne 34Ø).
Chaque résultat est mémorisé en A (F-1) (ligne 36Ø).

POUR JOUER :
On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe − et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole. Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.
L'appareil accepte, comme signes opératoires les symboles +, −, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.
Pour faire une autre opération, il faudra taper ESCAPE puis RUN.
Taper ESCAPE pour sortir du programme.

EXTENSIONS POSSIBLES :
On devra, tout d'abord, améliorer l'affichage du signe – (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres, on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

```
   10 REM CALCULATRICE
   20 DIM A(20)
   30 Z=0
   40 VDU23,224,255,255,255,255,255,255,
255,255
   50 D$=CHR$(224):C$=D$+D$+D$+D$+"      "
+D$+D$+" "+D$
   60 B$="14441333313121131314413312131
121411333314141 14131"
   70 F=1:L=0:COLOUR 128:CLS:COLOUR 2
   80 N=0:T=0:K=0:R$=""
   90 A$=GET$
  100 IF K=1 THEN N=0:R$="":K=0
  110 IF A$="=" THEN 290
  120 X=(1 AND A$="+")+(2 AND A$="-")+(3
 AND A$="*")+(4 AND A$="/")
  130 R$=R$+A$
  140 IF X<>0 THEN 290
  150 IF R$="-" THEN A$=R$:GOTO 230
  160 IF A$<"." OR A$>"9" OR A$="/" THEN
 90
  170 IF N=0 THEN CLS
  180 IF A$="." OR A$="-" THEN 230
  190 FOR I=1 TO 5
  200     S=3*VAL(MID$(B$,I+5*VAL(A$),1))
-2
  210    PRINT TAB(N,I+3);MID$(C$,S,3)
  220    NEXT I
  230 IF A$="-" THEN PRINT TAB(1,7);D$+D
$
  240 IF A$="." THEN PRINT TAB(N+1,8);D$
  250 N=N-4*(N<36)
  260 IF T<>0 THEN T=T+1:IF T<=LEN(R$) T
HEN 410
  270 IF K=1 AND L<>1 THEN 80
  280 T=0:GOTO 90
  290 IF R$="-" AND A(F)=0 THEN 150
  300 N=0:K=1
  310 IF (A$="*" OR A$="/") AND VAL(R$)=
0 THEN A(F)=1:GOTO 330
  320 A(F)=VAL(R$)
  330 IF X<>0 THEN Z=X
```

```
  340 ON Z GOSUB 430,450,470,490
  350 IF A$<>"=" THEN F=F+1 ELSE F=2
  360 A(F-1)=VAL(R$)
  370 IF VAL(R$)=0 THEN A(F-1)=A(F)
  380 R$=STR$(VAL(R$))
  390 IF LEFT$(R$,1)=" " THEN R$=MID$(R$
,2)
  400 T=1
  410 A$=MID$(R$,T,1)
  420 GOTO 170
  430 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):
L=1
  440 RETURN
  450 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):
A(F)=VAL(R$):L=1
  460 RETURN
  470 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):
A(F)=VAL(R$):L=1
  480 RETURN
  490 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):
A(F)=VAL(R$):L=1
  500 RETURN
>VDU12,3
```

Echiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : graphisme.
Difficulté du jeu : variable, selon utilisation.

LE JEU :
Le programme proposé vient en appoint du programme Echecs ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases (on pourra facilement l'étendre à 1Ø cases sur 1Ø cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H), et, verticalement, par des nombres (de 1 à 8).

LE PROGRAMME :
La ligne 6Ø imprime les lettres horizontales.
La boucle I (7Ø-13Ø) réalise l'impression, en utilisant la procédure **PROC-NORM,** qui positionne les couleurs à utiliser.

POUR JOUER :
L'ordre RUN provoque l'affichage du damier sur l'écran.
Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés rouges ou blancs. Les cases sont obtenues en largeur par une tabulation de la forme 3*J+2, J variant de 1 à 8, et en hauteur par 3*I+2.

EXTENSIONS POSSIBLES :
On pourra utiliser ce programme pour toute sorte de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```
   10 REM ECHIQUIER
   20 MODE 1
   30 CLS
   50 PRINT
   60 PRINT"    A   B   C   D   E   F   G   H"
   70 FOR I=1TO8
   80   FORJ=1TO3
   85     IF J=2 THEN PROCNORM (3,0):PRI
NT;I
   90     FORK=1TO4
  100        PROCNORM (0,3) :PRINT TAB((
K*6)+(I MOD 2)*3-3,I*3+J);"    "
  105        PROCNORM (0,1) :PRINT TAB((
K*6)+((I+1) MOD 2)*3-3,I*3+J)"    "
  110       NEXTK
  120     NEXTJ
  130   NEXTI
  140 PROCNORM(3,0)
  999 END
 1000  DEF PROCNORM(CL,FO)
 1010 COLOUR CL : COLOUR 128+FO
 1020 ENDPROC
>VDU12,3
```

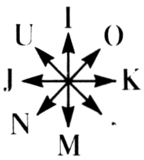
Zombies

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire. mais vous allez, vous aussi, essayer de les utiliser à votre profit. pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre. vont apparaître de façon aléatoire. les ZOMBIES (ce sont des Z). les trous noirs (des carrés jaunes et rouges) et le joueur (un O). Les ZOMBIES convergent vers le joueur pour essayer de le capturer. mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.
La partie se termine, soit avec la disparition du joueur. soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :
La position des ZOMBIES est définie par un tableau Z.
Les lignes 5Ø à 12Ø (boucle I) dessinent le cadre. les ZOMBIES et les trous noirs.
La ligne 13Ø définit la position initiale du joueur.
Une deuxième boucle I (17Ø-26Ø) réalise le déplacement du joueur.
Les lignes 28Ø à 34Ø contrôlent le déplacement du joueur.

POUR JOUER :
Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 8 touches ci-dessous et en veillant à ne pas sortir des limites du terrain.
Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie avant la fin, il suffira de taper ESCAPE.



EXTENSIONS POSSIBLES :
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.
On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```
   10 REM ZOMBIES
   20 DIM Z(10)
   30 MODE1:COLOUR 128:CLS
   40 COLOUR130:COLOUR 1
   50 FOR I=0 TO 21
   60   PRINT TAB(I+1,0);" "
   70   PRINT TAB(0,I);" ";TAB(23,I);" "
   80   IF RND(1)<.7 THEN PRINT TAB(I,RN
D(20)),"#"
   90   S=RND(14)
  100   Z(INT(I/3)+1)=5+S
  110   PRINT TAB(I+1,21);" "
  120   NEXT I:COLOUR 128:COLOUR 2
  130 K=RND(18)
  140 GOTO 320
  150 P=0
  160 COLOUR 2
  170 FOR I=1 TO 4
  180   IF Z(I)=0 THEN 260
  190   PRINT TAB(Z(I+4),Z(I));" "
  200   Z(I)=Z(I)-SGN(Z(I)-S)
  210   Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
  220   IF FNSCRN(Z(I+4),Z(I))=2 THEN Z(
I)=0:FOR Q=0 TO 10:SOUND 1,-1,130-Q*2,1:
NEXT Q:GOTO 240
  230   PRINT TAB(Z(I+4),Z(I));"Z":SOUND
 1,-1,150,2
  240   IF Z(I)=S AND Z(I+4)=K THEN PRIN
T TAB(0,25);"===  VOUS AVEZ PERDU...":EN
D
  250   P=P+Z(I)
  260   NEXT I
  270 IF P=0 THEN PRINT TAB(0,25);"BRAVO
. TOUS LES ZOMBIES SONT DETRUITS...":END
  280 K$=GET$
  290 PRINT TAB(K,S);" "
  300 K=K-(K$="O" OR K$="K" OR K$=",")+(
K$="U" OR K$="J" OR K$="N")
  310 S=S-(K$="N" OR K$="M" OR K$=",")+(
K$="U" OR K$="I" OR K$="O")
  320 COLOUR3:PRINT TAB(K,S);"O":SOUND 1
,-1,50,3
  330 IF K<=0 OR K>=23 OR S<=0 OR S>=21
THEN PRINT TAB(0,25);"DOMMAGE ...":END
  340 GOTO 150
  350 DEF FNSCRN(X,Y)
  360 =POINT(X*32,1023-Y*32)
>LOA
```

Le lézard d'Alice

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :
Les lignes 100 à 150 dessinent les verres et la procédure 350-430 dessine le lézard.
La boucle I (250-270) remplit les verres avec la quantité indiquée en ligne 230.
La variation de taille du lézard est définie par la ligne 290, avec contrôle par les lignes 300 et 310.
Un message gagnant sera affiché par la ligne 330 si l'on a trouvé la bonne taille.

POUR JOUER :
On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question COTE à laquelle on répond en tapant G ou D : on obtient l'affichage QUANTITE : on entre alors une quantité (de 0 à 130) et on frappe RETURN.
Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. Pour sortir, tapez ESCAPE.

EXTENSIONS POSSIBLES :
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire *Alice au Pays des Merveilles*, les idées n'y manquent pas.

```
   10I=0
   20J=3
   30REM LE LEZARD D'ALICE
   40CLS
   50D$="###########################"
   60L$="  ###   ###0#    #####     ####
#     ##      ## "
   70F$="         #    #####  ##   ##  "
   80N=12
   90A$="@"
  100COLOUR131:COLOUR 1:FOR I=1 TO 9
  110PRINT TAB(3,I+8);A$;TAB(10,I+8);A$;
TAB(13,I+8);A$;TAB(20,I+8);A$
  120NEXT
  130PROCLEZARD(N)
  140PRINT TAB(3,18);STRING$(8,A$)
  150PRINT TAB(13,18);STRING$(8,A$)
  160COLOUR128:PRINT TAB(4,22);"GAUCHE
  DROITE"
  170COLOUR 128:PRINT TAB(23,15);"COTE ?
 (G/D) : ";:K$=GET$
  180IF K$="D" THEN PRINT TAB(28,15);"DR
OIT     ":GOTO210
  190IF K$="G" THEN PRINT TAB(28,15);"GA
UCHE    ":GOTO210
  200VDU7:GOTO 170
  210U=4-10*(K$="D")
  220FOR I=1TO9:PRINT TAB(4,I+8);"
";TAB(14,I+8);"      ":NEXT I
  230INPUT TAB(23,18);"QUANTITE : "Q
  240 IF Q>130 THEN VDU7:GOTO 230
  250COLOUR 130:FOR I=1 TO Q/14.45+1
  260PRINT TAB(U,18-I);"      "
  270NEXT I
  280H=-4*(RND(1)<.5)-2
  290N=N+H*SQR(Q)
  300IF N<0 THEN N=0
  310IF N>26 THEN N=26
  320PROCLEZARD(N)
  330IF N=21 THEN PRINT TAB(0,25);"=== B
RAVO ...":END
  340GOTO 170
  350DEF PROCLEZARD(N)
  360COLOUR128
  370FOR I=1 TO 6
  380PRINT TAB(0,I);STRING$(40," ");TAB(
0,I);MID$(L$,8*I-7,8);
  390IF I=4 AND N<>0 THEN PRINTLEFT$(D$,
N);
  400PRINT TAB(N+8,I);MID$(F$,5*I-4,5)
  410NEXT
  420COLOUR130
  430ENDPROC
>VDU12,3
```

Chapeaux

REFERENCES :
Difficulté de programmation : assez facile.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

LE JEU :
L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange de chapeaux lors du croisement.

LE PROGRAMME :
Les lignes 20 à 40 contiennent le dessin des caractères graphiques nécessaires.
Les lignes 70 et 80 définissent, dans deux variables X$ et Y$, les dessins des deux personnages.
La ligne 220 permet, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 220 sert à repérer s'il y a eu échange ou non.
Le dessin des deux personnages est réalisé par une boucle I (120-210), et leur déplacement est réalisé par la boucle X (90-230).
Les lignes 260 à 290 déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :
Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et 0.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (a droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :
Il sera possible de prévoir un affichage couleur, avec une animation des personnages.
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```
   10 REM CHAPEAUX
   15 CLS
   20 VDU23,224,255,255,255,255,255,255,
255,255
   30 VDU23,225,0,0,0,0,0,255,255,0
   40 VDU 23,226,255,255,255,255,255,255
,255,0
   50 T=0
   60 D=10
   70 X$="332111131010030111113131011 01
101"
   80 Y$="233111131010030111113131011 01
101"
   90 FOR X=D TO 40-D
  100   Y=40-X
  110   L=1
  120   FOR I=1 TO LEN(X$) STEP 3
  130     PRINTTAB(X,L);" ";
  140     FORK=0TO2:PRINT CHR$(223+VAL(M
ID$(X$,I+K,1)));
  150       NEXT K:PRINT" "
  160     PRINTTAB(Y,L);" ";
  170     FORK=0TO2:PRINT CHR$(223+VAL(M
ID$(Y$,I+K,1)));
  180       NEXT K:PRINT" "
  190     L=L+1
  200     R$=INKEY$(1):IF R$="9" OR (R$>
="0" AND R$<="2") THEN 250
  210     NEXT I
  220   IF X=Y AND RND(1)<.5 THEN T=1:Z$
=X$:X$=Y$:Y$=Z$
  230   NEXT X
  240 PRINT TAB(0,25);"VOUS AVEZ OUBLIE
DE JOUER ?":END
  250  IF X<=Y THEN 210
  260 IF (R$="0" AND T=1) OR (R$="9" AND
 T=0) THEN PRINT TAB(0,25);"BRAVO... DRO
IT":END
  270 IF (R$="1" AND T=1) OR (R$="2" AND
 T=0) THEN PRINT TAB(0,25);"BRAVO... GAU
CHE":END
  280 IF R$="2" OR R$="1" THEN PRINT TAB
(0,25);"PERDU... GAUCHE":END
  290 PRINT TAB(0,25);"PERDU... DROIT":E
ND
>VDU 12,3
```

Othello

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu.
Le principe est que le joueur possédant les X doit pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).
Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :
Le tableau **SC** dimensionné à 2 à la ligne 3Ø contient les scores.
Les lignes 6Ø à 8Ø définissent les caractères graphiques (Case libre, joueur 1, joueur 2). On affiche le terrain aux lignes 1ØØ à 13Ø et les quatre premiers coups en 14Ø, 15Ø.
Le score est affiché grâce à la boucle **W** (17Ø-19Ø).
La ligne 2ØØ teste la fin de partie.
La ligne 28Ø permet le changement de joueur sans jeu.
Le déplacement est géré en 3ØØ-31Ø et contrôlé en 32Ø.
Le coup proprement dit est interprété aux lignes 36Ø-58Ø.
La boucle **P** (39Ø-53Ø) permet d'examiner les huit directions possibles, la variable **DT** contenant en fin de boucle le nombre de retournements effectués.

POUR JOUER :
Après le démarrage par **RUN,** le joueur qui a le trait peut déplacer son curseur grâce aux + flèches.
Il tape RETURN quand il est sur la case qu'il veut jouer. Si le coup est valide, l'Electron procédera au retournement des pions gagnés, comptabilise et affiche le nouveau score, et passe la main à l'adversaire. S'il est invalide, un **BIP** vous le signalera.
En cas d'impossibilité du jeu, il suffit de taper un slash (/) pour changer de main.

EXTENSIONS POSSIBLES :
Le plus intéressant est bien sûr d'imaginer un jeu-machine pour Electron.

```
10 REM OTHELLO
20 *FX4,1
30 DIMSC(2)
40 SC(1)=2:SC(2)=2
50 MODE1:COLOUR 128:CLS
60 VDU23,224,0,0,0,0,8,0,0,0
70 VDU23,225,0,73,28,54,119,54,28,73
80 VDU23,226,0,127,65,85,73,85,65,127
90 C1=5:L1=5:C2=21:L2=21
100 VDU28,2,22,22,2:COLOUR 131:CLS
110 VDU28,5,21,21,5:COLOUR 128:CLS
120 VDU26:COLOUR 3
130 FOR I=1 TO 8:PRINT TAB(3,2*I+4);I;
TAB(2*I+4,3);I:FOR J=1 TO 8:PRINT TAB(2*
I+4,2*J+4);CHR$(224):NEXT J,I
140   COLOUR 1:PRINT TAB(12,14);CHR$(2
25);TAB(14,12);CHR$(225)
150   COLOUR 2:PRINT TAB(12,12);CHR$(2
26);TAB(14,14);CHR$(226)
160   C=1:I=10:J=10
170   FOR W=1 TO 2
180     COLOUR W:PRINT TAB(28,8+W*3);"
JOUEUR ";CHR$(224+W);" ";SC(W);" "
190     NEXT W
200   IF SC(1)+SC(2)=64 THEN PRINT TAB
(0,28):END
210   COLOUR 128:COLOUR C:PRINT TAB(18
,24);"JOUEUR ";CHR$(224+C)
220   D=POINT(32*I+4,1019-32*J)
230   COLOUR D:COLOUR 128:PRINT TAB(I,
J);CHR$(224+D)
240   K=INKEY(0):IF K<>-1 THEN 270
250   COLOUR 3-D:COLOUR 128+D:PRINT TA
B(I,J);CHR$(224+D)
260   GOTO230
270   IF K=13 THEN 360
280   IF K=47 THEN C=3-C:GOTO 170
290   X=I:Y=J
300   I=I-2*(K=137)+2*(K=136)
310   J=J-2*(K=138)+2*(K=139)
320   IF I<C1 OR I>C2 OR J<L1 OR J>L2
THEN I=X:J=Y:SOUND 1,-1,100,2:GOTO 250
330   IF D<>0 COLOUR D ELSE COLOUR 3
340   COLOUR 128:PRINT TAB(X,Y);CHR$(2
24+D)
350   GOTO 220
360   IF D<>0 THEN SOUND 1,-1,100,2:GO
TO 250
370   COLOUR 3
```

```
380   DT=0
390   FOR P=1 TO 8
400     G=-(P>5)+(P>1 AND P<5)
410     F=-(P>3 AND P<7)+(P=8 OR P<3)
420     F=2*F:G=2*G:DP=0
430     I1=I:J1=J
440     I1=I1+F:J1=J1+G:D1=POINT(32*I1
+4,1019-32*J1)
450     IF D1=0 OR D1=3 THEN 530
460     IF D1=3-C THEN DP=DP+1:GOTO 44
0
470     IF DP=0 THEN 530
480     FOR K=1 TO DP
490       I1=I1-F:J1=J1-G
500       DT=DT+1
510       COLOUR C:COLOUR 128:PRINT TA
B(I1,J1);CHR$(224+C)
520       NEXT K
530     NEXT P
540   IF DT=0 THEN SOUND 1,-1,100,2:GO
TO250
550   SC(C)=SC(C)+DT+1:SC(3-C)=SC(3-C)
-DT
560   COLOUR C:COLOUR 128:PRINT TAB(I,
J);CHR$(224+C)
570   C=3-C
580   GOTO 170
590   END
600   DEF PROCFEN(X,Y,T,C)
610   VDU28,X,Y+T-1,X+T-1,Y
620   COLOUR 128+C:CLS
630   COLOUR 128:VDU26
640   ENDPROC
650   COLOUR 128:COLOUR 3:CLS
>
```

Achevé d'imprimer par l'imprimerie Tardy Quercy S.A. à Bourges
Dépôt légal : Février 1985 - N° d'Imprimeur : 12152 - N° d'Édition : 2-86595-212-1

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- Ce livre vous donne-t-il toute satisfaction?

____

____

____

____

____

- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?

____

____

____

____

____

Comment avez-vous eu connaissance de ce livre?

☐ publicité
☐ catalogue
☐ boutique micro
☐ autres
☐ cadeau
☐ librairie
☐ exposition

Avez-vous déjà acquis des livres PSI?

lesquels? ______________________________

____

qu'en pensez-vous? ______________________________

____

____

Nom ____________ Prénom ____________ Age ______
Adresse ______________________________
Profession ______________________________
Centre d'intérêt ______________________________

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

Apprendre en se distrayant, tel est l'objectif de ce livre. Ces 102 programmes de jeu vous guideront progressivement dans l'exploration du BASIC de l'**Électron**. Niveau par niveau, vous assimilerez de nouvelles instructions pour arriver à la maîtrise de votre ordinateur. Chaque niveau commence par une présentation pédagogique du jeu d'instructions utilisé. Chaque jeu est décrit et les programmes sont analysés ligne par ligne afin d'en faciliter la modification. Devenez vite le chef d'orchestre de votre **Électron**!

DES LIVRES POUR VOTRE ORDINATEUR

P.S.I.

ISBN: 2.86595.212.6
120 FF

Couverture GIBRAT

102 PROGRAMMES POUR ÉLECTRON

PAR JACQUES DECONCHAT

PROGRAMMES
P.S.I.